Budget-Aware User Satisfaction Maximization on Service Provisioning in Mobile Edge Computing

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

Abstract—Mobile Edge Computing (MEC) promises to provide mobile users with delay-sensitive services at the edge of network, and each user service request usually is associated with a Service Function Chain (SFC) requirement that consists of Virtualized Network Functions (VNFs) in order. The satisfaction of a user on his requested service is heavily impacted by the service reliability. In this article, we study user satisfaction on services provided by an MEC network through introducing a submodular function based metric to measure user satisfaction. We first formulate a novel user satisfaction problem with the aim to maximize the accumulative user satisfaction, assuming that all available computing resource in the MEC network can be used for service reliability enhancement. We show that the problem is NP-hard, and devise an approximation algorithm with a provable approximation ratio for it. We then consider the problem under a given computing resource budget constraint, for which we devise an approximation algorithm with a provable approximation ratio, at the expense of moderate budget violations. We finally evaluate the performance of the proposed algorithms through experimental simulations. Simulation results demonstrate that the proposed algorithms outperform the comparison baseline algorithms, improving the performance by more 16.1% in comparison with the baseline algorithms.

Index Terms—Mobile Edge Computing (MEC), user service satisfaction, Virtualized Network Function (VNF), Service Function Chain (SFC), VNF instance placement, reliable virtual service provisioning, budget-aware generalized assignment problem, approximation algorithms, resource allocation and optimization

1 Introduction

Fueled by the surging demands of mobile users, Mobile Edge Computing (MEC) is emerging as a promising technology to provide delay-sensitive services to mobile users at the edge of networks to shorten service delays in the evolution to 5 G and beyond 5 G networks [4], [10], [11], [16], [24], [29]. With this concept, Network Function Virtualization (NFV) is another emerging paradigm to endow network services with agility and flexibility in terms of designing, instantiation, and management [1], [5], [21], [45]. In this vein, NFV provides a fertile ground to exploit the substantial benefits of the Internet of Things (IoT) by deploying Virtualized Network Function (VNF) instances on virtual machines to support network services [7], [9], [33]. Furthermore, a Service Function Chain (SFC) that consists of VNFs in order is able to provision complicated and integrated network services [12].

Providing users with satisfaction services while maintaining SFC enforcements of users is an important requirement in service provisioning of MEC networks [15], [22], [37]. User satisfaction on services usually is proportional to service reliability. However, SFCs in NFV-enabled MEC networks are subject to failures, because the malfunction of any single VNF in a SFC will disrupt the reliability of the whole chain [6], and the reliability of an SFC substantially decreases with the increase on the chain length [44]. A common remedy for dealing with the failures is to adopt redundancy, i.e., deploy multiple backup VNF instances for each primary VNF instance in advance, and all backup VNF instances remain idle until the primary one fails [43].

- *Jing Li is with the School of Computing, The Australian National University, Canberra, ACT 2601, Australia. E-mail: u6013763@anu.edu.au.*
- *Weifa Liang and Xiaohua Jia are with the Department of Computer Science, City University of Hong Kong, Hong Kong, China. E-mail: {weifa.liang, csjia}@cityu.edu.hk.*
- *Wenzheng Xu is with the School of Computer Science, Sichuan University, Chengdu 610017, China. E-mail: wenzheng.xu@scu.edu.cn.*
- *Zichuan Xu is with the School of Software, Dalian University of Technology, Dalian 116024, China. E-mail: z.xu@dlut.edu.cn.*
- *Albert Y. Zomaya is with the School of Computer Science, The University of Sydney, Sydney, NSW 2006, Australia. E-mail: albert.zomaya@sydney.edu.au.*
- *Song Guo is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China. E-mail: song.guo@polyu.edu.hk.*

Manuscript received 9 December 2021; revised 24 June 2022; accepted 6 September 2022. Date of publication 9 September 2022; date of current version 3 November 2023.
The work of Jing Li and Weifa Liang was supported in part by Australian Research Council under its Discovery Project Scheme with under Grant DP200101985. The work of Weifa Liang was also supported in part by the City University of Hong Kong under Grant 9380137/CS. The work of Wenzheng Xu was supported in part by NSFC under Grant 61602330. The work of Zichuan Xu was supported in part by the National Natural Science Foundation of China under Grant 61802048 and the "Xinghai Scholar Program" in Dalian University of Technology, China. The work of Xiaohua Jia was supported in part by the Research Grants Council of Hong Kong with under Grant CityU 9042813. The work of Song Guo was supported in part by funding from Hong Kong RGC Research Impact Fund (RIF) with under Grant R5060-19, in part by General Research Fund (GRF) with 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.
(Corresponding author: Weifa Liang.)
Digital Object Identifier no. 10.1109/TMC.2022.3205427

Existing studies [6], [13], [14], [17], [19], [20], [26], [27], [28], [35], [42], [44] assumed that each VNF instance has an expected failure probability, and each user request is associated with a service reliability requirement as a hard threshold to determine whether its user is satisfied, i.e., if the expected service reliability of the provisioned SFC service to a user is no less than the specified reliability requirement, the user will satisfy the service; otherwise, the user will not satisfy the service. However, it is very difficult to predict the failure probability of a VNF instance precisely [37], and the failure probability of a VNF instance varies over time [36]. It thus is unrealistic to measure a user satisfaction by simply using the expected service reliability of each user individually. Instead, it is desirable to maximize the accumulative user satisfaction on services provided by an MEC network through the deployment of backups of VNF instances to improve the service reliability of all services. Furthermore, the network service provider of an MEC network usually has a given computing resource budget for the improvement of user satisfaction [2], [32]. To maximize user satisfaction on the services provided by an MEC network, it imposes the following challenges. How to quantitatively model user service satisfaction? how to allocate the limited computing resource in the MEC network for different users to maximize the accumulative user satisfaction on services? how to develop an efficient scheduling algorithm for maximizing user satisfaction under a given computing resource budget? In the rest of this paper, we will address these challenges.

The novelty of the work in this paper lies in that we introduce a submodular function based metric to measure user satisfaction on SFC-enabled service provisioning in MEC environments. We formulate a novel user satisfaction problem with and without the computing resource budget constraints, and develop the very first performance-guaranteed approximation algorithms for it.

The main contributions of this paper are as follows. We first formulate a novel user satisfaction problem for services provided by an MEC network through enhancing the SFC reliability of requested services by users, and show that the problem is NP-hard. We then devise an approximation algorithm with a provable approximation ratio for the problem without the computing resource budget constraint, and analyze the approximation ratio and time complexity of the proposed algorithm. We also develop an approximation algorithm with a provable approximation ratio for the problem with a given computing resource budget, and the approximate solution obtained is at the expense of a moderate budget violation. We finally evaluate the performance of the proposed algorithms through simulations. Simulation results demonstrate that the proposed algorithms outperform the comparison baseline algorithms, improving the performance by more than 16.1% in comparison with that of the baseline algorithms.

The rest of the paper is organized as follows. Section 2 summarizes the related work on reliability-aware service provisioning in MEC. Section 3 introduces the system model and problem definition. Section 4 shows the NP-hardness of the defined problem. Section 5 devises an approximation algorithm for the user satisfaction problem without the computing resource budget constraint. Section 6 devises an approximation algorithm for the user satisfaction problem with the computing resource budget constraint. Section 7 evaluates the proposed algorithms empirically, and Section 8 concludes the paper.

2 Related Work

There have been several efforts on reliability-aware VNF service provisioning in MEC networks recently. For example, He et al. [13] took both reliabilities of VNFs and backup servers into account through distinguishing the importance of different VNFs. They proposed efficient heuristic algorithms to minimize the weighted unavailability of VNFs. Huang et al. [14] investigated a VNF service provisioning problem by deploying backup VNF instances on different cloudlets to meet reliability requirements of users, and proposed different approximation algorithms under different assumptions. Kanizo et al. [17] studied a backup optimization problem for VNFs with the aim of achieving high reliability. They also took resource sharing into consideration in order to reduce resource consumptions significantly. Li et al. [19], [20] jointly considered the reliability of VNF instances and the reliability of cloudlets in an MEC network, and proposed online algorithms for a problem to maximize the accumulative revenue. Yala et al. [42] developed an efficient heuristic algorithm to address a VNF placement problem to maximize the reliability of provisioning services while minimizing the service latency through incorporating both edge clouds and central clouds. However, none of these studies ever considered user satisfaction on their demanded services with SFC requirements. Xu et al. [41] recently investigated machine-learning driven NVF markets to maximize the social welfare of all players so that all players have incentives to participate in the activities of the market.

There are studies on reliability-aware SFC service provisioning in MEC networks. For example, Fan et al. [6] addressed a reliability-aware SFC mapping problem to protect provisioned services. They aimed to maximize the acceptance ratio of requests while minimizing the resource consumption. Lin [28] considered a reliability-aware SFC provisioning problem, and devised a randomized algorithm and a heuristic algorithm with the aim to maximize the number of requests admitted. Liang et al. [26], [27] investigated a service reliability augmentation problem, considering latency constraint on timely synchronizations between primary and backup VNF instances when an update occurs. Qu et al. [35] proposed an efficient resource sharing strategy among backup VNF instances to provision reliable SFC services under both the bandwidth and computing resource constraints. They developed a heuristic for VNF assignment to meet reliability requirements of users. Wang [39] merged multiple SFCs into a service function graph (SFG) based on VNF instance sharing and function dependency. They mapped the SFG to network nodes, and proposed a node-ranking algorithm with centrality and reliability (NRCR) for VNF backup deployments to achieve resource efficiency, while meeting SFC reliability requirements. Zhang et al. [44] aimed to minimize the computing resource consumption of backups for meeting the SFC reliability demands of users, and developed an efficient heuristic algorithm for the problem, considering heterogeneous computing resource demands of different VNF instances. However, these mentioned studies distinguished a user satisfaction from dissatisfaction by simply comparing the expected service reliability against a given reliability requirement. Shang et al. [36] developed a self-adapting backup scheme for SFC services with the aim to minimize the total backup cost. They deployed one static backup for each primary VNF instance, and then deployed dynamic backups to guarantee reliable SFC service provisioning in an online manner. Thiruvasagam [38] proposed heuristic algorithms for efficient SFC placements with the aim to minimize the implementation cost, considering both service reliability and service delays. In addition to deploying a dedicated backup for each VNF, they also deployed virtual Monitoring Functions (vMFs) to monitor VNFs and mitigate service degradation. However, the computing resource budget constraint was not considered in their study.

In our previous studies [23], [25], we studied user satisfaction problem for delay-sensitive IoT services provided jointly by an MEC network and a remote cloud, where a user satisfaction was assumed to be inversely proportional to the extra service delay beyond each user's delay threshold (i.e., the tolerance degree beyond the specified delay threshold). Each request only demands a single service while each service has only one service instance, and no SFC requirement is associated with any service request. Meeting a service request with SFC enforcement while meeting service reliability is much more challenging. The approaches and techniques adopted in [23], [25] are not applicable to the problem in this paper. We here focus on user service satisfaction improvement through augmenting SFC service reliability with and without a given computing resource budget constraint with the aim to maximize the accumulative user satisfaction, by deploying backups for primary VNF instances in SFCs for admitted user requests.

3 Preliminary

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

3.1 System Model

Consider an MEC network as an undirected graph $G = (V, E)$, where V is the set of nodes and E is the set of links between nodes. Each node $v \in V$ is an Access Point (AP) co-located with a cloudlet. The AP and its co-located cloudlet are connected through a high-speed optical cable, and the communication delay between them is negligible [30], [31]. Denote by $C(v)$ the residual computing resource of cloudlet v, assuming that the primary VNF instances of SFCs of all admitted service requests have been placed in the system. Primary VNF instances are the active VNF instances, while all backup VNF instances remain idle until their primary ones fail. Denote by σ_v the cost of computing resource on cloudlet v per unit.

Denote by $\mathcal{F}$ the set of different types of VNFs offered by the network service provider in G, $c(f)$ the demanded computing resource of an instance of VNF $f \in \mathcal{F}$, and $r(f)$ the expected reliability of VNF f in any cloudlet with $0 < r(f) < 1$. Without loss of generality, we assume that the

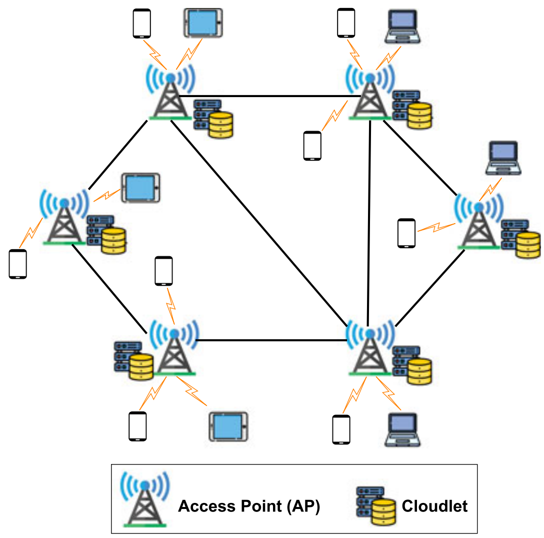


Fig. 1. An illustrative example of an MEC network consisting of 6 APs, and each AP is co-located with a cloudlet.

computing resource capacity of any cloudlet can accommodate at least one backup VNF instance of any type, i.e., $\min_{v \in V}\{C(v)\} \geq \max_{f \in \mathcal{F}}\{c(f)\}$. Fig. 1 is an illustrative example of an MEC network.

3.2 User Service Requests With Service Reliability

Consider a set U of user requests. Let SC_j be the requested SFC of request $u_j \in U$. Denote by $f_{i,j} \in \mathcal{F}$ the ith VNF of SFC SC_j. Let $\mathbb{S}$ be the set of backup VNF instances deployed in all cloudlets. Denote by $n_{i,j}(\mathbb{S})$ the number of primary and backup VNF instances deployed for VNF $f_{i,j}$ with the backups from $\mathbb{S}$, and $n_{i,j}(\mathbb{S}) \geq 1$. We further assume that each primary VNF instance can have at most K backups, i.e., $n_{i,j}(\mathbb{S}) - 1 \leq K$, where K (≥ 1) is a given positive integer.

The reliability $R_{\mathbb{S}}(f_{i,j})$ of VNF $f_{i,j}$ with the backups from set $\mathbb{S}$ is calculated as follows.

$$R_{\mathbb{S}}(f_{i,j}) = 1 - (1 - r(f_{i,j}))^{n_{i,j}(\mathbb{S})}, \tag{1}$$

where $r(f_{i,j})$ is the reliability of VNF $f_{i,j}$.

The reliability $R_{\mathbb{S}}(u_j)$ of the SFC SC_j of user request u_j with backups from set $\mathbb{S}$ can be calculated as follows.

$$R_{\mathbb{S}}(u_j) = \prod_{f_{i,j} \in SC_j} R_{\mathbb{S}}(f_{i,j}). \tag{2}$$

Denote by $R_{init}(u_j)$ the reliability of SC_j by deploying primary VNF instances without any backup deployment, then

$$R_{init}(u_j) = \prod_{f_{i,j} \in SC_j} r(f_{i,j}). \tag{3}$$

3.3 Submodular Function and Utility Gain

Let Ω be a finite set, *a submodular function* is a set function $h : 2^{\Omega} \rightarrow \mathbb{R}^{\geq 0}$, where $\mathbb{R}^{\geq 0}$ is the non-negative real number set. Function $h(\cdot)$ is a submodular function if for every $\mathbb{A}, \mathbb{B} \subseteq \Omega$ with $\mathbb{A} \subseteq \mathbb{B}$ and every $a \in \Omega \setminus \mathbb{B}$, $h(\mathbb{A} \cup \{a\}) - h(\mathbb{A}) \geq h(\mathbb{B} \cup \{a\}) - h(\mathbb{B})$.

A user satisfaction on a service increases with the increase of the service reliability, while the increasing rate of the satisfaction becomes slower with the improvement of the service reliability. Such a user satisfaction can be modelled by a submodular function. In this paper, we thus adopt a logarithmic function that is a submodular function to model each user satisfaction of using services with SFC requirements in an MEC network.

Given a backup set $\mathbb{S}$, the user satisfaction of user u_j by placing primary and backup VNF instances for his SFC requirement is modelled by a submodular utility function $\log_2 \frac{R_{\mathbb{S}}(u_j)}{\lambda}$, where λ is a given constant with $0 < \lambda < \min_{u_j \in U}\{R_{init}(u_j)\}$. The *utility gain* by placing the backups from set $\mathbb{S}$ for user u_j thus is

$$\rho_j(\mathbb{S}) = \log_2 \frac{R_{\mathbb{S}}(u_j)}{\lambda} - \log_2 \frac{R_{init}(u_j)}{\lambda} = \log_2 \frac{R_{\mathbb{S}}(u_j)}{R_{init}(u_j)}. \quad (4)$$

The *total utility gain* by placing the backups from set $\mathbb{S}$ for all user requests in U thus is

$$g(\mathbb{S}) = \sum_{u_j \in U} \rho_j(\mathbb{S}) = \sum_{u_j \in U} \log_2 \frac{R_{\mathbb{S}}(u_j)}{R_{init}(u_j)}. \quad (5)$$

We then have the following important lemma for the defined utility function $g(\cdot)$.

Lemma 1. *The defined utility gain function $g(\cdot)$ in Eq. (5) is a submodular function.*

Proof. Let $\mathbb{A}$ and $\mathbb{B}$ be two sets of backups deployed in the MEC network with $\mathbb{A} \subseteq \mathbb{B} \subset \mathbb{S}$. We have $n_{i,j}(\mathbb{A}) \le n_{i,j}(\mathbb{B})$, i.e., the number of primary and backup VNF instances of each VNF for each user request with set $\mathbb{A}$ is no larger than that with $\mathbb{B}$, respectively.

Let $a \notin \mathbb{B}$ be a backup to be deployed for VNF $f_{i,'j'}$, i.e., $n_{i,'j'}(\mathbb{A} \cup \{a\}) = n_{i,'j'}(\mathbb{A}) + 1$ and $n_{i,'j'}(\mathbb{B} \cup \{a\}) = n_{i,'j'}(\mathbb{B}) + 1$. We then have

$$\begin{aligned} & g(\mathbb{A} \cup \{a\}) - g(\mathbb{A}) = \rho_{j'}(\mathbb{A} \cup \{a\}) - \rho_{j'}(\mathbb{A}) && (6)\\ =& \log_2 \frac{R_{\mathbb{A}\cup\{a\}}(u_{j'})}{R_{init}(u_{j'})} - \log_2 \frac{R_{\mathbb{A}}(u_{j'})}{R_{init}(u_{j'})} \\ =& \log_2 R_{\mathbb{A}\cup\{a\}}(u_{j'}) - \log_2 R_{\mathbb{A}}(u_{j'}) \\ =& \sum_{f_{i,j'} \in SC_{j'}} \log_2 R_{\mathbb{A}\cup\{a\}}(f_{i,j'}) - \sum_{f_{i,j'} \in SC_{j'}} \log_2 R_{\mathbb{A}}(f_{i,j'}) \\ =& \log_2 R_{\mathbb{A}\cup\{a\}}(f_{i,'j'}) - \log_2 R_{\mathbb{A}}(f_{i,'j'}) && (7)\\ =& \log_2 \frac{1-(1-r(f_{i,'j'}))^{n_{i,'j'}(\mathbb{A})+1}}{1-(1-r(f_{i,'j'}))^{n_{i,'j'}(\mathbb{A})}}, && (8) \end{aligned}$$

where Eq. (6) holds because deploying backup a only increases the utility gain from request $u_{j'}$ while the utility gain of the other requests does not change. Meanwhile, Eq. (7) holds because deploying backup a only increases the service reliability of VNF $f_{i,'j'}$, while the reliability of the other VNFs in $SC_{j'}$ does not change. Eq (8) holds due to that $n_{i,'j'}(\mathbb{A} \cup \{a\}) = n_{i,'j'}(\mathbb{A}) + 1$.

For the sake of convenience, we replace $(1 - r(f_{i,'j'}))$ by $m \in (0,1)$ and $n_{i,'j'}(\mathbb{A})$ by $n(\mathbb{A})$ (≥ 1). Then,

$$g(\mathbb{A} \cup \{a\}) - g(\mathbb{A}) = \log_2 \frac{1 - m^{n(\mathbb{A})+1}}{1 - m^{n(\mathbb{A})}}, \quad (9)$$

similarly,

$$g(\mathbb{B} \cup \{a\}) - g(\mathbb{B}) = \log_2 \frac{1 - m^{n(\mathbb{B})+1}}{1 - m^{n(\mathbb{B})}}. \quad (10)$$

Because $\log_2(x)$ is a monotonically increasing function with $x > 0$, to compare $(g(\mathbb{A} \cup \{a\}) - g(\mathbb{A}))$ and $(g(\mathbb{B} \cup \{a\}) - g(\mathbb{B}))$, we instead compare $\frac{1-m^{n(\mathbb{A})+1}}{1-m^{n(\mathbb{A})}}$ and $\frac{1-m^{n(\mathbb{B})+1}}{1-m^{n(\mathbb{B})}}$. That is, if $\frac{1-m^{n(\mathbb{A})+1}}{1-m^{n(\mathbb{A})}} - \frac{1-m^{n(\mathbb{B})+1}}{1-m^{n(\mathbb{B})}} \ge 0$, then $g(\mathbb{A} \cup \{a\}) - g(\mathbb{A}) \ge g(\mathbb{B} \cup \{a\}) - g(\mathbb{B})$. The detailed proof is given as follows.

$$\begin{aligned} & \frac{1 - m^{n(\mathbb{A})+1}}{1 - m^{n(\mathbb{A})}} - \frac{1 - m^{n(\mathbb{B})+1}}{1 - m^{n(\mathbb{B})}} \\ =& \frac{(1-m)\cdot(m^{n(\mathbb{A})} - m^{n(\mathbb{B})})}{(1 - m^{n(\mathbb{A})})\cdot(1 - m^{n(\mathbb{B})})} \ge 0, \quad (11) \end{aligned}$$

where Ineq. (11) holds because $m = 1 - r(f_{i,'j'}) \in (0,1)$, and we have $1 \le n(\mathbb{A}) \le n(\mathbb{B})$ due to $\mathbb{A} \subseteq \mathbb{B} \subset \mathbb{S}$. Thus, $g(\mathbb{A} \cup \{a\}) - g(\mathbb{A}) \ge g(\mathbb{B} \cup \{a\}) - g(\mathbb{B})$.

The lemma then follows. □

3.4 Problem Definition

Given an MEC network $G = (V, E)$, a positive integer K, a given computing resource budget $B > 0$, a set of cloudlets V with each having computing capacity, a set U of admitted user service requests with each request having an SFC requirement, assuming the primary VNF instances of the SFCs of these admitted service requests have already been placed in the network. The *user satisfaction problem* in G is to maximize the total utility gain, by deploying up to K backups for each primary VNF instance in cloudlets for admitted requests, subject to the computing capacity on each cloudlet in G and the given budget B.

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

4 NP-Hardness of the Problem

In this section, we show that the defined problem is NP-hard as follows.

Theorem 1. *The user satisfaction problem in an MEC network $G = (V, E)$ is NP-hard.*

Proof. The NP-hardness of the user satisfaction problem is shown by a reduction from a well-known NP-hard problem - the knapsack problem [34] as follows.

Given M items and a bin with capacity of W, each item i has a weight w_i and a profit p_i with $1 \le i \le M$, the knapsack problem is to maximize the total profit by packing as many items as possible into the bin, subject to the bin capacity.

We consider a special case of the user satisfaction problem without the computing resource budget constraint, where there is only one cloudlet in the MEC network with computing capacity W after the placement of primary VNF instances of all admitted requests. We treat the cloudlet as a bin with capacity W. For each request $u_j \in U$, we assume that its requested SFC consists of only a single VNF f_j. Recall that we can deploy up to K backups for VNF f_j on the cloudlet. We treat each potential

TABLE 1
Table of Symbols

Notations	Descriptions
$G = (V, E)$	An MEC network as an undirected graph $G = (V, E)$, where V is the set of nodes and E is the set of links between nodes
$C(v)$	Residual computing resource of cloudlet v, assuming that the primary VNF instances of all admitted requests have been placed
σ_v	Cost of one unit computing resource on cloudlet v
σ_{max} and σ_{min}	The maximum and minimum costs per unit of computing resource among cloudlets in the MEC network
$\mathcal{F}$	Set of different types of VNFs offered by the network service provider in G
$c(f)$	Demanded computing resource of an instance of VNF $f \in \mathcal{F}$
$r(f)$	Expected reliability of VNF f with $0 < r(f) < 1$
U and u_j	A set of user service requests and a user service request
SC_j and $f_{i,j}$	The requested SFC of request $u_j \in U$ and the ith VNF of the SFC SC_j
$\mathbb{S}$	Set of backup VNF instances deployed for all users in all cloudlets
$n_{i,j}(\mathbb{S})$	The number of primary and backup VNF instances deployed for VNF $f_{i,j}$ with the backups from $\mathbb{S}$, and $n_{i,j}(\mathbb{S}) \geq 1$
K	Each primary VNF instance can have at most K backups with $K \geq 1$
$R_{\mathbb{S}}(f_{i,j})$	The reliability of the VNF $f_{i,j}$ with the backups from set $\mathbb{S}$
$R_{\mathbb{S}}(u_j)$	The reliability of the SFC SC_j of user request u_j with the backups from set $\mathbb{S}$
$R_{init}(u_j)$	The reliability of the SFC SC_j by deploying primary VNF instances without any backup deployment
$\rho_j(\mathbb{S})$	The utility gain by placing the backups from set $\mathbb{S}$ for user u_j
$g(\mathbb{S})$	Total utility gain by placing the backups from set $\mathbb{S}$ for all user requests in U
$d_{i,j,k}$	The kth backup of VNF $f_{i,j}$
$p(d_{i,j,k})$	The utility gain by deploying the kth backup $d_{i,j,k}$ of VNF $f_{i,j}$
$x^v_{i,j,k}$	Binary decision variable, where $x^v_{i,j,k} = 1$ indicates that backup $d_{i,j,k}$ is placed to cloudlet v, and $x^v_{i,j,k} = 0$ otherwise
B	Given computing resource budget
$\mathcal{V}$	A cloudlet with the computing capacity $C(\mathcal{V})$ $(= \frac{B}{\sigma_{min}})$
$y_{i,j,k}$	Binary decision variable, where $y_{i,j,k} = 1$ indicates that backup $d_{i,j,k}$ is placed to cloudlet $\mathcal{V}$, and $y_{i,j,k} = 0$ otherwise
$\lvert SC\rvert_{\max}$	The maximum length of an SFC among all requests
$\mathbb{S}_1$ and $\mathbb{S}_2$	Backup set $\mathbb{S}$ is partitioned into two disjoint subsets, i.e., $\mathbb{S} = \mathbb{S}_1 \cup \mathbb{S}_2$ and $\mathbb{S}_1 \cap \mathbb{S}_2 = \emptyset$
$\mu_1, \mu_2, \ldots, \mu_{\lvert\mathbb{S}\rvert}$	Sorted backups in $\mathbb{S}$ in non-decreasing order of computing resource consumption
α and ϵ	Constants with $0 < \alpha \leq 1$ and $0 < \epsilon \leq 1$

backup of f_j as an item, and there are K items for f_j: $d_{j,1}, d_{j,2}, \ldots, d_{j,K}$, where the weight $w(d_{j,k})$ of item $d_{j,k}$ is the amount of computing resource consumed by a VNF instance of f_j with $1 \leq k \leq K$, i.e., $w(d_{j,k}) = c(f_j)$. The profit $p(d_{j,k})$ of item $d_{j,k}$ is the utility gain of deploying the kth backup VNF instance for f_j, assuming that its first $(k-1)$ backup VNF instances have already been placed, i.e., $p(d_{j,k}) = \rho_j(\{d_{j,1}, \ldots, d_{j,k-1}, d_{j,k}\}) - \rho_j(\{d_{j,1}, \ldots, d_{j,k-1}\})$, and $p(d_{j,1}) = \rho_j(\{d_{j,1}\})$ initially. There are M $(= |U| \cdot K)$ items in total for all admitted requests. This special user satisfaction problem is to maximize the total utility gain by deploying as many backup VNF instances as possible for all admitted requests on the cloudlet, subject to computing capacity W on the cloudlet.

It can be seen that all items $d_{j,k}$ derived from VNF f_j have the same weight $c(f_j)$, while the profit of item $d_{j,k}$ decreases with the increase of the value of k, since $g(\cdot)$ is a submodular function by Lemma 1, i.e., $w(d_{j,k}) = w(d_{j,k+1})$ while $p(d_{j,k}) > p(d_{j,k+1})$. Thus, when packing items into the cloudlet for request u_j, items $d_{j,k}$ is placed to the cloudlet in increasing order of k, where $1 \leq k \leq K$. Let $\mathbb{S}$ be the set of chosen items to be placed to the cloudlet and k_j items are placed for VNF f_j, $\forall u_j \in U$. Since each item corresponds to a potential backup, in the following we show that the total profit gain by packing as many items in $\mathbb{S}$ as possible to the bin is equal to the total utility gain $g(\mathbb{S})$ by deploying as many backups in $\mathbb{S}$ as possible. The total profit gain of packing items in $\mathbb{S}$ is

$$\begin{aligned}
&\sum_{u_j \in U} (p(d_{j,1}) + p(d_{j,2}) + \ldots + p(d_{j,k_j})) \\
&= \sum_{u_j \in U} (\rho_j(\{d_{j,1}\}) + \rho_j(\{d_{j,1}, d_{j,2}\}) - \rho_j(\{d_{j,1}\}) \\
&\quad + \ldots + \rho_j(\{d_{j,1}, \ldots, d_{j,k_j}\}) - \rho_j(\{d_{j,1}, \ldots, d_{j,k_j - 1}\})) \\
&= \sum_{u_j \in U} \rho_j(\{d_{j,1}, \ldots, d_{j,k}\}) = \sum_{u_j \in U} \rho_j(\mathbb{S}) = g(\mathbb{S}). \qquad (12)
\end{aligned}$$

It can be seen that if there is a solution to this special user satisfaction problem, there is a solution to the knapsack problem and the reduction is polynomial. Hence, the user satisfaction problem is NP-hard, because the knapsack problem is NP-hard [34]. □

5 Approximation Algorithm for the Problem Without the Computing Resource Budget

In this section, we deal with the user satisfaction problem without the computing resource budget constraint, and all available resource in an MEC network can be used for backup VNF placements. We first provide an integer linear programming (ILP) formulation for the problem, which will serve as the benchmark purpose when the problem size is small. Otherwise, we devise a performance-guaranteed approximation algorithm for the problem. We then analyze the approximation ratio and time complexity of the proposed approximation algorithm.

5.1 ILP Formulation

Since up to K backups of each VNF $f_{i,j}$ can be placed to cloudlets to improve the service reliability of service request u_j, let $d_{i,j,1}, d_{i,j,2}, \ldots, d_{i,j,K}$ be the K potential backups of VNF $f_{i,j}$, and let $c(d_{i,j,k})$ be the amount of computing resource demanded by backup $d_{i,j,k}$ with $1 \leq k \leq K$, then

$$c(d_{i,j,k}) = c(f_{i,j}). \tag{13}$$

It can be seen that the total utility gain $g(\mathbb{S})$ of deploying backups in set $\mathbb{S}$ by Eq. (5) is

$$\begin{aligned} g(\mathbb{S}) &= \sum_{u_j \in U} \log_2 \frac{R_{\mathbb{S}}(u_j)}{R_{init}(u_j)} = \sum_{u_j \in U} \log_2 \prod_{f_{i,j} \in SC_j} \frac{R_{\mathbb{S}}(f_{i,j})}{r(f_{i,j})} \\ &= \sum_{u_j \in U} \sum_{f_{i,j} \in SC_j} \log_2 \frac{R_{\mathbb{S}}(f_{i,j})}{r(f_{i,j})} \\ &= \sum_{u_j \in U} \sum_{f_{i,j} \in SC_j} \log_2 \frac{1-(1-r(f_{i,j}))^{n_{i,j}(\mathbb{S})}}{r(f_{i,j})}. \end{aligned} \tag{14}$$

Assuming that the first $(k-1)$ backups of VNF $f_{i,j}$ have been deployed already. By Eq. (14), the utility gain $p(d_{i,j,k})$ by deploying the kth backup $d_{i,j,k}$ of VNF $f_{i,j}$ is

$$\begin{aligned} p(d_{i,j,k}) &= \log_2 \frac{1-(1-r(f_{i,j}))^{k+1}}{r(f_{i,j})} - \log_2 \frac{1-(1-r(f_{i,j}))^{k}}{r(f_{i,j})} \\ &= \log_2 \frac{1-(1-r(f_{i,j}))^{k+1}}{1-(1-r(f_{i,j}))^{k}}. \end{aligned} \tag{15}$$

An integer linear programming (ILP) formulation for the user satisfaction problem is given as follows.

Let $x^v_{i,j,k}$ be a binary decision variable, where $x^v_{i,j,k} = 1$ indicates that backup $d_{i,j,k}$ is placed to cloudlet v, and $x^v_{i,j,k} = 0$ otherwise. The problem is to

$$\text{maximize} \quad \sum_{v \in V} \sum_{u_j \in U} \sum_{f_{i,j} \in SC_j} \sum_{1 \leq k \leq K} p(d_{i,j,k}) \cdot x^v_{i,j,k} \tag{16}$$

subject to the following constraints.

Eq. (13) and (15),

$$\sum_{u_j \in U} \sum_{f_{i,j} \in SC_j} \sum_{1 \leq k \leq K} c(d_{i,j,k}) \cdot x^v_{i,j,k} \leq C(v), \quad \forall v \in V \tag{17}$$

$$\sum_{v \in V} x^v_{i,j,k} \leq 1, \quad \forall u_j \in U, \forall f_{i,j} \in SC_j, \quad \forall k \in [1, K] \tag{18}$$

$$x^v_{i,j,k} \in \{0,1\}, \forall u_j \in U, \forall f_{i,j} \in SC_j, \forall k \in [1,K], \forall v \in V, \tag{19}$$

where the objective Function (16) is the accumulative utility gain of all admitted requests in U. Constraint (17) is the computing capacity constraint on each cloudlet. Constraint (18) indicates that a VNF backup can be deployed to at most one cloudlet, and it cannot be deployed across multiple cloudltes. Constraint (19) says that $x^v_{i,j,k}$ is a binary decision variable, indicating whether backup $d_{i,j,k}$ is deployed in cloudlet v or not.

5.2 Approximation Algorithm

We devise an approximation algorithm for the problem by reducing it to the maximum-profit generalized assignment problem (GAP), and an approximate solution to the latter in turn returns an approximate solution to the former.

Given M items and N bins, each item i has a weight w_i and a profit p_i with $1 \leq i \leq M$, while each bin j has capacity W_j with $1 \leq j \leq N$. *The maximum-profit GAP* is to maximize the total profit by packing as many items as possible to N bins, subject to the capacity on each bin. The maximum-profit GAP is NP-hard, and there is an approximation algorithm for it [3].

The reduction of the problem of concern to the maximum-profit GAP is as follows. Each cloudlet $v \in V$ is a bin with capacity $C(v)$, and there are $|V|$ bins. Each backup $d_{i,j,k}$ is an item with a weight $c(d_{i,j,k})$ by Eq. (13) and a profit $p(d_{i,j,k})$ by Eq. (15). It can be seen that if there is a solution to the maximum-profit GAP, there exists a solution to the problem of concern. The proposed approximation algorithm for the user satisfaction problem without the computing resource budget constraint is given in `Algorithm` 1.

Algorithm 1. Algorithm for the User Satisfaction Problem Without the Computing Resource Budget Constraint

Input: An MEC network $G = (V, E)$ and a set U of admitted requests with each request having an SFC requirement.

Output: A backup deployment strategy to maximize the total utility gain of all admitted requests without the computing resource budget constraint.

for each request $u_j \in U$ **do**
  for each VNF $f_{i,j} \in SC_j$ **do**
    for each $k \in [1, K]$ **do**
      Let $d_{i,j,k}$ be a potential backup of VNF $f_{i,j}$;
      $c(d_{i,j,k}) \leftarrow c(f_{i,j})$;
      $p(d_{i,j,k}) \leftarrow \log_2 \frac{1-(1-r(f_{i,j}))^{k+1}}{1-(1-r(f_{i,j}))^{k}}$;
    end for
  end for
end for
Construct an instance of the maximum-profit GAP, where each cloudlet v corresponds to a bin with capacity $C(v)$, and each backup $d_{i,j,k}$ corresponds to an item with weight $c(d_{i,j,k})$ and profit $p(d_{i,j,k})$;
Find a solution $\mathbb{S}$ for the maximum-profit GAP by invoking the approximation algorithm in [3];
return A feasible solution for the problem of concern is derived from solution $\mathbb{S}$ directly.

5.3 Algorithm Analysis

In the following, we first analyze the important properties of the ILP solution. We then analyze the approximation ratio and time complexity of `Algorithm` 1.

Lemma 2. *Given a VNF $f_{i,j}$, the utility gain $p(d_{i,j,k})$ of its backups $d_{i,j,k}$ decreases with the increase on the value of k, i.e., $p(d_{i,j,k}) > p(d_{i,j,k'})$ with $1 \leq k < k' \leq K$.*

Proof. For the sake of convenience, we here substitute $(1 - r(f_{i,j}))$ with $m \in (0,1)$. The difference of the utility gain between $d_{i,j,k}$ and $d_{i,j,k'}$ with $k < k'$ is

$$p(d_{i,j,k})-p(d_{i,j,k'})=\log_2\frac{1-m^{k+1}}{1-m^k}-\log_2\frac{1-m^{k'+1}}{1-m^{k'}}. \quad (20)$$

Due to the fact that $\log_2(x)$ is a monotonically increasing function with $x > 0$, to compare $p(d_{i,j,k})$ and $p(d_{i,j,k'})$, we instead compare $\frac{1-m^{k+1}}{1-m^k}$ and $\frac{1-m^{k'+1}}{1-m^{k'}}$. That is, if $\frac{1-m^{k+1}}{1-m^k}-\frac{1-m^{k'+1}}{1-m^{k'}} > 0$, then $p(d_{i,j,k}) > p(d_{i,j,k'})$.

We then have

$$\frac{1-m^{k+1}}{1-m^k}-\frac{1-m^{k'+1}}{1-m^{k'}}=\frac{(1-m)\cdot(m^k-m^{k'})}{(1-m^k)\cdot(1-m^{k'})} > 0, \quad (21)$$

because $0 < m < 1$ and $k < k'$. □

Lemma 3. *For a VNF $f_{i,j}$, its backups $d_{i,j,k}$ with $1 \le k \le K$ are deployed in cloudlets in increasing order of k, i.e., if backup $d_{i,j,k+1}$ is deployed in a cloudlet, then backup $d_{i,j,k}$ must have already been deployed in one of the cloudlets in the MEC network.*

Proof. We show the claim by contradiction. Assuming that backup $d_{i,j,k+1}$ is deployed in a cloudlet while $d_{i,j,k}$ has not been deployed in any cloudlet yet. As each backup of VNF $f_{i,j}$ has the same computing resource consumption $c(f_{i,j})$, the utility gain due to the deployment of the VNF backup decreases with the increase of the value of k by Lemma 2. Thus, backup $d_{i,j,k+1}$ can be replaced by $d_{i,j,k}$ and a larger utility gain can be obtained without violating computing resource capacity of any cloudlet. This results in a contradiction. □

Lemma 4. *Given a set $\mathbb{S}$ of NFV backup instances delivered by the ILP (16), then the ILP solution (16) is $g(\mathbb{S})$.*

Proof. Recall that $(n_{i,j}(\mathbb{S})-1)$ is the number of backups deployed for VNF $f_{i,j}$ from backup set $\mathbb{S}$. By Lemma 3, for VNF $f_{i,j}$, its backups $d_{i,j,1}, d_{i,j,2}, \ldots, d_{i,j,n_{i,j}(\mathbb{S})-1}$ are deployed at different cloudlets in G, while the rest backups $d_{i,j,n_{i,j}(\mathbb{S})}, \ldots, d_{i,j,K}$ are not deployed at any cloudlet. Then, the total utility gain of deploying these $(n_{i,j}(\mathbb{S})-1)$ backups of VNF $f_{i,j}$ is

$$\begin{aligned}&p(d_{i,j,1})+p(d_{i,j,2})+\ldots+p(d_{i,j,n_{i,j}(\mathbb{S})-1})\\ &=\log_2\frac{1-(1-r(f_{i,j}))^2}{r(f_{i,j})}+\ldots+\log_2\frac{1-(1-r(f_{i,j}))^{n_{i,j}(\mathbb{S})}}{1-(1-r(f_{i,j}))^{n_{i,j}(\mathbb{S})-1}}\\ &=\log_2\frac{1-(1-r(f_{i,j}))^{n_{i,j}(\mathbb{S})}}{r(f_{i,j})}.\end{aligned} \quad (22)$$

The total utility gain of deploying backups in $\mathbb{S}$ thus is

$$\begin{aligned}&\sum_{u_j\in U}\sum_{f_{i,j}\in SC_j}(p(d_{i,j,1})+\ldots+p(d_{i,j,n_{i,j}(\mathbb{S})-1}))\\ &=\sum_{u_j\in U}\sum_{f_{i,j}\in SC_j}\log_2\frac{1-(1-r(f_{i,j}))^{n_{i,j}(\mathbb{S})}}{r(f_{i,j})}\\ &=\sum_{u_j\in U}\log_2\frac{\prod_{f_{i,j}\in SC_j}(1-(1-r(f_{i,j}))^{n_{i,j}(\mathbb{S})})}{\prod_{f_{i,j}\in SC_j}r(f_{i,j})}\\ &=\sum_{u_j\in U}\log_2\frac{R_{\mathbb{S}}(u_j)}{R_{init}(u_j)}=g(\mathbb{S}).\end{aligned} \quad (23)$$

The lemma then follows. □

Theorem 2. *Given an MEC network $G=(V,E)$, a set V of cloudlets with computing capacity, a set U of admitted user service requests with each request having an SFC requirement, assuming the SFCs have been placed in the network, there is a $\frac{1}{2+\alpha}$-approximation algorithm,* `Algorithm 1`*, for the user satisfaction problem without the computing resource budget constraint. The algorithm takes $O(|V|\cdot|U|\cdot|SC|_{\max}\cdot K\cdot\log\frac{1}{\alpha}+\frac{|V|}{\alpha^4})$ time, where $|SC|_{\max}$ is the maximum length of an SFC among all requests, α is a constant with $0 < \alpha \le 1$, and K is the maximum number of backups deployed for any VNF.*

Proof. The approximation ratio of `Algorithm 1` is $\frac{1}{2+\alpha}$, which is derived from the approximation algorithm in [3]. The time complexity of `Algorithm 1` is dominated by the running time of the approximation algorithm from [3], which is $O(|V|\cdot|U|\cdot|SC|_{\max}\cdot K\cdot\log\frac{1}{\alpha}+\frac{|V|}{\alpha^4})$. □

6 Approximation Algorithm for the Problem With Computing Resource Budget

In this section, we study the user satisfaction problem with a given computing resource budget B. That is, the total cost of amounts of computing resource consumed for supporting backup VNF instance placements of all admitted requests is no greater than a given value B. For the sake of convenience, denote by **P1** and **P2** the problem of concern and another optimization problem that is derived from problem **P1**, respectively. The general strategy for tackling the user satisfaction problem **P1** is as follows.

We first formulate an integer linear programming (ILP) solution to problem **P2**, and then obtain an approximate solution to problem **P2**. We then derive an approximate solution to problem **P1** based on the approximate solution to problem **P2**, at the expense of a moderate budget violation.

Due to the computing resource budget constraint on problem **P1**, it can be seen that the total amount of computing resources of all cloudlets in G for service reliability augmentation of admitted requests is no more than $\frac{B}{\sigma_{min}}$, where B is the computing resource budget and $\sigma_{min}=\{\sigma_v \mid v\in V\}$ is the minimum cost per unit of computing resource among all cloudlets.

Problem **P2** is defined as follows. Consider a very special MEC network that consists of only a single cloudlet $\mathcal{V}$ with computing capacity $C(\mathcal{V})$ $(=\frac{B}{\sigma_{min}})$ and a set U of admitted user requests with each having an SFC requirement, assuming that the primary VNF instances of each request have already been placed in this special MEC network. Problem **P2** then is to maximize the accumulative user satisfaction by deploying up to K backups for each primary VNF instance in the SFC of each request, subject to the computing capacity on cloudlet $\mathcal{V}$ in the MEC network.

If the computing resource budget constraint on problem **P1** can be ignored, the problem actually makes use of all available computing resources of all cloudlets in the MEC network G for user satisfaction enhancement, and the total cost of computing resource consumption is no less than a given computing resource budget B, i.e., $\sum_{v\in V}\sigma_v\cdot C(v)\ge B$, where σ_v is the cost per unit computing resource in cloudlet v.

In the rest of our discussion, we assume that $\sum_{v\in V}\sigma_{min}\cdot C(v)\ge B$. Thus, $\sum_{v\in V}C(v)\ge\frac{B}{\sigma_{min}}=C(\mathcal{V})$, i.e., the total available computing resource of all cloudlets in problem **P1** is no less than the computing capacity of the single cloudlet $\mathcal{V}$ in problem **P2**. The computing resource budget constraint B restricts the amount of computing resource for user satisfaction enhancements, which also implies that the network service provider need reserving adequate amount of computing resource for new service request admissions.

6.1 ILP Formulation for Problem P2

We formulate an integer linear programming (ILP) solution to problem **P2**. Similar to the ILP formulation (16), let $\{d_{i,j,1}, d_{i,j,2}, \ldots, d_{i,j,K}\}$ be the set of potential K backups of each VNF $f_{i,j}$. Recall that the amount of computing resource consumed by backup $d_{i,j,k}$ is $c(d_{i,j,k})$ following Eq. (13), and the utility gain of backup $d_{i,j,k}$ is $p(d_{i,j,k})$ by Eq. (15).

Let $y_{i,j,k}$ be a binary decision variable, where $y_{i,j,k} = 1$ indicates that backup $d_{i,j,k}$ is placed to cloudlet $\mathcal{V}$ for VNF $f_{i,j}$; and $y_{i,j,k} = 0$ otherwise. The ILP formulation for problem **P2** is given as follows.

$$\textbf{P2}: \text{Maximize} \sum_{u_j \in U} \sum_{f_{i,j} \in SC_j} \sum_{1 \le k \le K} p(d_{i,j,k}) \cdot y_{i,j,k}, \tag{24}$$

subject to the following constraints.

Eq. (13) and (15),

$$\sum_{u_j \in U} \sum_{f_{i,j} \in SC_j} \sum_{1 \le k \le K} c(d_{i,j,k}) \cdot y_{i,j,k} \le C(\mathcal{V}), \tag{25}$$

$$y_{i,j,k} \in \{0,1\}, \ \forall u_j \in U, \forall f_{i,j} \in SC_j, \forall k \in [1, K], \tag{26}$$

where the objective Function (24) is the accumulated utility gain of admitted requests. Constraint (25) is the computing capacity constraint on cloudlet $\mathcal{V}$. Constraint (26) says that $y_{i,j,k}$ is a binary decision variable, indicating whether backup $d_{i,j,k}$ for VNF $f_{i,j}$ is deployed in cloudlet $\mathcal{V}$.

It can be seen that problem **P2** is the classic knapsack problem, and there is an approximation algorithm for it [18].

6.2 Approximation Algorithm

We deal with the user satisfaction problem with a given computing resource budget constraint (problem **P1**), by developing an approximate solution that is derived from an approximate solution of problem **P2** as follows.

We construct an instance of problem **P2** based on problem **P1**. Because problem **P2** is a knapsack problem, we obtain an approximate solution (i.e., a backup set $\mathbb{S}$) for it, by applying an approximation algorithm in [18]. We then derive an approximate solution to problem **P1** by an approximation algorithm, which essentially redistributes the backups in $\mathbb{S}$ for problem **P1** to different cloudlets in G iteratively. We also partition set $\mathbb{S}$ into two disjoint subsets $\mathbb{S}_1$ and $\mathbb{S}_2$ to avoid computing capacity violation of cloudlets. The procedure of redistribution and partitioning of backups is presented as follows.

We sort the backups in $\mathbb{S}$ in non-decreasing order of their computing resource consumption, and we sort cloudlets in V in non-decreasing order of their unit computing resource cost σ_v, too. Without loss of generality, we assume that the sorted backup sequence is $\mu_1, \mu_2, \ldots, \mu_{|\mathbb{S}|}$, and the sorted cloudlet sequence is $v_1, v_2, \ldots, v_{|V|}$, respectively.

Initially, we deploy backups to the first cloudlet v_1 one by one, until after deploying backup μ_l with $1 \le l \le |\mathbb{S}|$, the computing resource of cloudlet v_1 is running out (i.e., no residual computing resource in cloudlet v_1 can accommodate the next backup), or the computing capacity of cloudlet v_1 is violated if the next backup is added. If deploying μ_l results in no computing resource violation, μ_l is added to set $\mathbb{S}_1$; otherwise, μ_l is added to set $\mathbb{S}_2$, and the violation will be dealt later. We then deploy the rest backups $\mu_{l+1}, \ldots, \mu_{|\mathbb{S}|}$ one by one to the next cloudlet v_2, and so on. This procedure continues until all backups in $\mathbb{S}$ are deployed. Because the computing capacity on each cloudlet can be violated, and the total amount of computing resource for problem **P1** is assumed to be no less than that for problem **P2**, i.e., $\sum_{v \in V} C(v) \ge C(\mathcal{V}) = \frac{B}{\sigma_{min}}$, then all backups in $\mathbb{S}$ can be deployed in cloudlets of the MEC network for problem **P1**.

Algorithm 2. Algorithm for the User Satisfaction Problem With the Computing Resource Budget Constraint

Input: An MEC network $G = (V, E)$, a set U of admitted requests with SFC requirements and a computing resource budget B.

Output: A backup deployment strategy to maximize the total utility gain under the computing resource budget constraint.

```
Let 𝒱 be the only cloudlet with capacity of C(𝒱) = B/σ_min in a special MEC network for problem P2;
for each request u_j ∈ U do
  for each VNF f_{i,j} ∈ SC_j do
    for each k ∈ [1, K] do
      Let d_{i,j,k} be a potential backup of VNF f_{i,j};
      c(d_{i,j,k}) ← c(f_{i,j});
      p(d_{i,j,k}) ← log_2 [(1-(1-r(f_{i,j}))^{k+1}) / (1-(1-r(f_{i,j}))^k)];
    end for
  end for
end for
Find an approximate solution 𝕊 to problem P2, by invoking the approximation algorithm in [18];
Sort the backup VNF instances in 𝕊 in non-decreasing order of computing resource consumption;
Sort cloudlets in V in non-decreasing order of computing resource cost σ_v per unit;
𝕊_1 ← ∅; 𝕊_2 ← ∅; t ← 1;
for each μ_l ∈ 𝕊 with the sorted order do
  Deploy backup μ_l on cloudlet v_t, and update the residual computing resource of cloudlet v_t;
  if the computing capacity C(v_t) of v_t is violated then
    𝕊_2 ← 𝕊_2 ∪ {μ_l}; t ← t + 1;
  else if the computing resource of v_t is running out then
    𝕊_1 ← 𝕊_1 ∪ {μ_l}; t ← t + 1;
  else
    𝕊_1 ← 𝕊_1 ∪ {μ_l};
  end if ;
end for
if g(𝕊_1) > g(𝕊_2) then
  return 𝕊_1 and g(𝕊_1);
else
  return 𝕊_2 and g(𝕊_2);
end if .
```

Set $\mathbb{S}$ now is partitioned into two disjoint subsets $\mathbb{S}_1$ and $\mathbb{S}_2$, i.e., $\mathbb{S} = \mathbb{S}_1 \cup \mathbb{S}_2$ and $\mathbb{S}_1 \cap \mathbb{S}_2 = \emptyset$. A backup μ_l is added to set $\mathbb{S}_1$ when no computing capacity constraint is violated by deploying μ_l. For each cloudlet $v \in V$, at most one backup is added to set $\mathbb{S}_2$, because cloudlet v will not be considered for further backup deployment if its capacity is violated by deploying μ_l, and it is assumed that each cloudlet can accommodate at least one backup of any VNF. Then, deploying the backups in either $\mathbb{S}_1$ or $\mathbb{S}_2$ in cloudlets will not result in any computing resource violation. The proposed approximation algorithm for problem **P1** then chooses the one with a larger utility gain between the two sets $\mathbb{S}_1$ and $\mathbb{S}_2$ as the solution to the problem. The detailed algorithm for the user satisfaction problem is given in `Algorithm` 2.

6.3 Algorithm Analysis

The rest is to show the correctness of the proposed algorithms, and analyze the approximation ratio and time complexity of `Algorithm` 2. The computing resource budget violation of the solution delivered by `Algorithm` 2 is also shown to be bounded.

Lemma 5. *Given a set $\mathbb{S}$ of VNF backup instances delivered by the ILP (24) for problem **P2**, then the value of the ILP solution (24) is $g(\mathbb{S})$, i.e., the total utility gain of deploying the backups from $\mathbb{S}$.*

The proof is similar to the proof of Lemma 4, and omitted.

Lemma 6. *Given a set $\mathbb{S}$ of backup VNF instances for problem **P2**, the total cost of deploying the backups in $\mathbb{S}$ to cloudlets for problem **P1** in G is no more than $\frac{\sigma_{max}}{\sigma_{min}} \cdot B$, where B is the computing resource budget in problem **P1**, σ_{max} and σ_{min} are the maximum and minimum costs per unit of computing resource among cloudlets in G, respectively.*

Proof. Because the capacity of cloudlet $\mathcal{V}$ is $\frac{B}{\sigma_{min}}$, the amount of computing resource consumed by backups in $\mathbb{S}$ is at most $\frac{B}{\sigma_{min}}$. The total cost of deploying the backups in $\mathbb{S}$ to cloudlets for problem **P1** is no more than $\frac{\sigma_{max}}{\sigma_{min}} \cdot B$. □

Theorem 3. *Given an MEC network $G = (V, E)$, a set V of cloudlets with computing capacity, a set U of admitted user service requests with each request having an SFC requirement, and a given computing resource budget B on G, assuming that the primary VNF instances of SFCs have already been placed in the network, there is a $(\frac{1-\epsilon}{2})$-approximation algorithm, `Algorithm` 2, for the user satisfaction problem under a given computing resource budget constraint, at the expense of the computing resource budget violation upper bounded by $(\frac{\sigma_{max}}{\sigma_{min}} - 1)$. The algorithm takes $O(\mathcal{N} \cdot \log \frac{1}{\epsilon} + \frac{1}{\epsilon^4} + \mathcal{N} \cdot \log \mathcal{N} + |V| \cdot \log |V|)$ time, where ϵ is a given constant with $0 < \epsilon < 1$, $\mathcal{N} = |U| \cdot |SC|_{\max} \cdot K$, $|SC|_{\max}$ is the maximum length of an SFC, K is the maximum number of deployed backups for a VNF, and σ_{max} and σ_{min} are the maximum and minimum costs per unit of computing resource among all cloudlets, respectively.*

Proof. Let OPT_1 and OPT_2 be the optimal solutions to problems **P1** and **P2**, respectively. In problem **P2**, the computing resource budget constraint is ignored, and there is only a single cloudlet, the computing capacity of which is assumed to be no less than the total amount of computing resources of all cloudlets that can be used for service satisfaction augmentation in problem **P1** with the computing resource budget B. Then

$$OPT_2 \geq OPT_1. \tag{27}$$

Recall that $\mathbb{S}$ is the set of backup VNF instances delivered by the approximation algorithm for problem **P2** [18]. Denote by $\mathcal{A}_2(\mathbb{S})$ the total utility gain by deploying the backups in set $\mathbb{S}$ for problem **P2**. As the approximation ratio of the approximation algorithm for problem **P2** is $(1-\epsilon)$ [18], where ϵ is a constant with $0 < \epsilon < 1$. We then have

$$\mathcal{A}_2(\mathbb{S}) \geq (1-\epsilon) \cdot OPT_2 \geq (1-\epsilon) \cdot OPT_1. \tag{28}$$

Recall that set $\mathbb{S}$ is further partitioned into two disjoint subsets $\mathbb{S}_1$ and $\mathbb{S}_2$. Let $\mathcal{A}_1(\mathbb{S})$, $\mathcal{A}_1(\mathbb{S}_1)$, and $\mathcal{A}_1(\mathbb{S}_2)$ be the total utility gains by deploying backups from $\mathbb{S}$, $\mathbb{S}_1$, and $\mathbb{S}_2$ on cloudlets in the MEC network G for problem **P1**, respectively, then the utility gains of deploying backups in $\mathbb{S}$ for either problem **P1** or problem **P2** are the same if the computing resource budget constraint on problem **P1** is ignored, i.e., $\mathcal{A}_1(\mathbb{S}) = \mathcal{A}_2(\mathbb{S}) = g(\mathbb{S})$, then

$$\mathcal{A}_1(\mathbb{S}) = \mathcal{A}_2(\mathbb{S}) \geq (1-\epsilon) \cdot OPT_1. \tag{29}$$

The set with larger utility gain between $\mathbb{S}_1$ and $\mathbb{S}_2$ is chosen as the solution of problem **P1**. Then

$$\max\{\mathcal{A}_1(\mathbb{S}_1), \mathcal{A}_1(\mathbb{S}_2)\} \geq \frac{1}{2} \cdot \mathcal{A}_1(\mathbb{S}) \geq \frac{1-\epsilon}{2} \cdot OPT_1. \tag{30}$$

Since set $\mathbb{S}_1$ consists of backups without any computing capacity violation, each cloudlet is assigned with at most one backup in $\mathbb{S}_2$, and it is assumed that each cloudlet can accommodate at least one backup of any VNF, the deployment of the backups in $\mathbb{S}_1$ or $\mathbb{S}_2$ to cloudlets in G for problem **P1** will not result in any computing resource violation. By Lemma 6, the total cost of deploying backups in $\mathbb{S}$ for problem **P1** is no more than $\frac{\sigma_{max}}{\sigma_{min}} \cdot B$. The computing resource budget constraint on deploying backups in $\mathbb{S}_1$ or $\mathbb{S}_2$ thus is violated by no more than a factor of $(\frac{\sigma_{max}}{\sigma_{min}} - 1)$.

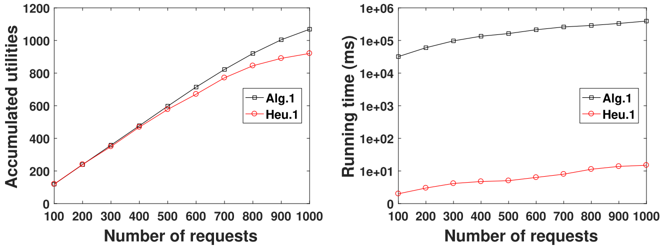
The time complexity of `Algorithm` 2 is analyzed as follows. The time complexity of the approximation algorithm [18] for problem **P2** is $O(\mathcal{N} \cdot \log \frac{1}{\epsilon} + \frac{1}{\epsilon^4})$, where $\mathcal{N} = |U| \cdot |SC|_{\max} \cdot K$ is the total number of potential backups for all admitted requests in U. Sorting the backups in $\mathbb{S}$ takes $O(\mathcal{N} \cdot \log \mathcal{N})$ time, and sorting the cloudlets takes $O(|V| \cdot \log |V|)$ time. The time complexity of `Algorithm` 2 thus is $O(\mathcal{N} \cdot \log \frac{1}{\epsilon} + \frac{1}{\epsilon^4} + \mathcal{N} \cdot \log \mathcal{N} + |V| \cdot \log |V|)$. □

7 Performance Evaluation

In this section, we evaluate the performance of the proposed algorithms through experimental simulation, and we also investigate the impact of important parameters on the performance of the proposed algorithms.

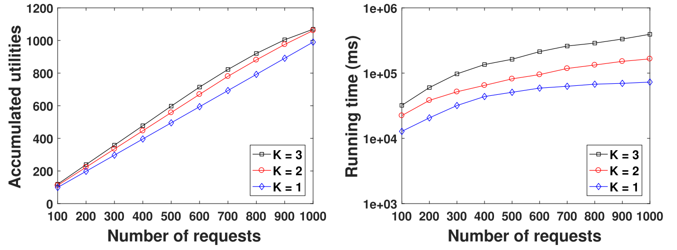
7.1 Experimental Environment Setting

We consider an MEC network $G = (V, E)$ consisting of 200 APs, and each AP is co-located with a cloudlet, where each network instance is generated by the widely used tool GT-



(a) The accumulated utilities (b) The running times

Fig. 2. Performance of different algorithms for the user satisfaction problem without the computing resource budget constraint.



(a) The accumulated utilities (b) The running times

Fig. 3. The impact of K on the performance of algorithm `Alg.1`.

ITM [8]. The computing capacity of each cloudlet is randomly drawn from 4,000 MHz to 12,000 MHz [26], [27]. Assume that there are 20 different types of VNFs [26], [27] offered by the network service provider with each having a reliability between 0.8 and 0.9 [44], and the computing resource consumption of a VNF instance is randomly drawn from 100 MHz to 200 MHz [14]. The computing resource cost of each cloudlet is randomly drawn from \$0.02 to \$0.03 per MHz [40]. The number of different types of SFCs is set as 30, and the length of each of them varies from 3 to 7 [30], [31]. For each request, a random SFC from the preset SFCs is requested, and its primary VNF instances are randomly deployed in cloudlets in the generated MEC network. Each primary VNF instance is assumed to be equipped with up to 3 backups, i.e., K is set at 3. The computing resource budget is set as $\$10,000$. Parameters α in `Algorithm` 1 and ϵ in `Algorithm` 2 are set as 0.5, respectively. The value in each figure is the mean of the results out of 30 MEC instances with the same size. The running time of each algorithm is obtained by a desktop with a 3.7 GHz 8-core Intel i7 CPU and 16 GB RAM. These parameters are adopted as the default settings unless otherwise specified.

To evaluate the proposed algorithm `Algorithm` 1, referred to as `Alg.1`, we introduce a heuristic algorithm `Heu.1`. For each request, it first deploys a backup for each VNF of its SFC one by one, and each such backup is deployed in a randomly chosen cloudlet with sufficient residual computing resource. If each primary VNF instance has one backup already, it then deploys the second backup of each primary VNF instance similarly. This procedure continues until either each VNF in the SFC has been equipped with K backups or there is no sufficient residual computing resource for further backup deployments.

To evaluate the proposed algorithm `Algorithm` 2, referred to as `Alg.2`, we propose a heuristic algorithm `Heu.2`, which deals with requests one by one. For each request, it first deploys a backup for each VNF of its SFC. Among cloudlets with sufficient residual computing resource, each backup is deployed to the cloudlet with the minimum computing resource cost. If each primary VNF instance of the SFC has been equipped with a backup, it then deploys the second backup for each VNF by the similar manner. This procedure continues until either there is no residual budget or each primary VNF instance has been equipped with K backups already, or there is insufficient computing resource for further backup deployments.

7.2 Algorithm Performance for the User Satisfaction Problem Without Computing Resource Budget Constraint

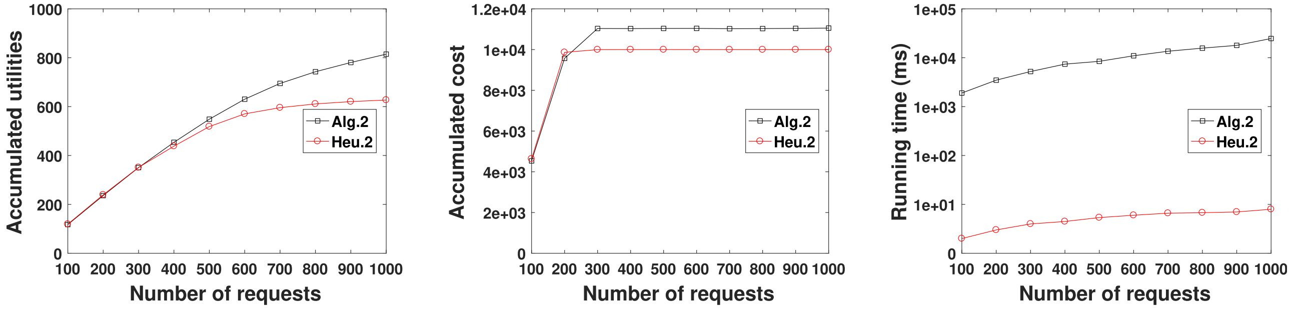
We first studied the performance of algorithm `Alg.1` against algorithm `Heu.1`, by varying the number of requests from 100 to 1,000. Fig. 2 depicts the accumulative utilities and running times of the comparison algorithms for the user satisfaction problem without the computing resource budget constraint. It can be seen from Fig. 2a that algorithm `Alg.1` outperforms algorithm `Heu.1` by at least 16.1% in terms of performance when there are 1,000 admitted requests. This is due to the fact that algorithm `Alg.1` better utilizes the computing resource to maximize the accumulative utility of all admitted requests.

We then evaluated the parameter K on the performance of algorithm `Alg.1` as each primary VNF instance can have up to K backups. Fig. 3 shows the accumulated utilities and running times of algorithm `Alg.1` by varying the number of requests when $K = 1, 2$, and 3, respectively. As depicted by Fig. 3a, algorithm `Alg.1` when $K = 3$ outperforms itself when $K = 1$ in all cases. Specifically, when there are 100 requests, the accumulative utility gain delivered by algorithm `Alg.1` when $K = 3$ is 17.9% more than that by itself when $K = 1$. When the number of requests reaches 1,000, the accumulative utility gain achieved by algorithm `Alg.1` when $K = 1$ is 92.6% of that by itself when $K = 3$. The rationale behind is that with a larger K, more backups of each primary VNF instance can be deployed, thereby improving the accumulative user satisfaction.

7.3 Algorithm Performance for the User Satisfaction Problem With the Computing Resource Budget Constraint

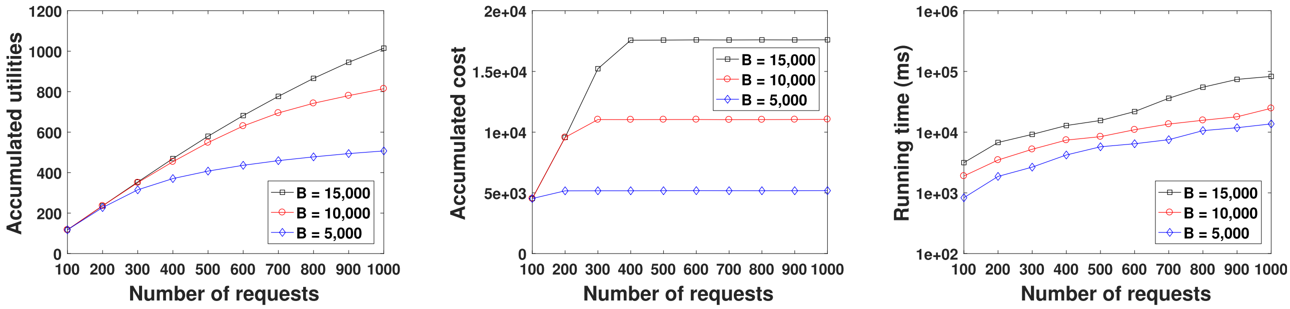
The rest is to study the performance of the proposed algorithm for the problem under a given computing resource budget B constraint.

We first investigated the performance of algorithm `Alg.2` against algorithm `Heu.2`, by varying the number of requests from 100 to 1,000. Fig. 4 plots the accumulated utilities, accumulated costs and running times of the two mentioned algorithms for the user satisfaction problem with the computing resource budget constraint. From Fig. 4a, it can be seen that algorithm `Alg.2` outperforms algorithm `Heu.2` by 30.9% with 1,000 requests. Recall that the computing resource budget is $\$10,000$, Fig. 4b shows that the computing resource budget constraint is violated by 10.5% with 1,000 requests. This is because algorithm `Alg.2` establishes an efficient backup deployment strategy to maximize the accumulative user satisfaction.



(a) The accumulated utilities (b) The accumulated costs (c) The running times

Fig. 4. Performance of different algorithms for the user satisfaction problem with the computing resource budget constraint.



(a) The accumulated utilities (b) The accumulated costs (c) The running times

Fig. 5. The impact of the budget B on the performance of algorithm `Alg.2`.

We then studied the impact of the computing resource budget B on the performance of algorithm `Alg.2`. Fig. 5 depicts the accumulated utilities, accumulated costs and running times of algorithm `Alg.2` by varying the computing resource budget B at 5,000, 10,000, and 15,000, respectively. Fig. 5a demonstrates that the performance of algorithm `Alg.2` when $B = 5,000$ is 49.8% of itself when $B = 15,000$. This is because more VNF backups can be deployed to gain more utilities with a larger computing resource budget. As shown by Fig. 5b, with 1,000 requests and $B = 5,000$, `Alg.2` violates the computing resource budget constraint by no more than 3.7%. In contrast, with 1,000 requests and $B = 15,000$, the computing resource budget constraint violation ratio by `Alg.2` is no more than 17.3%. The justification is that with a small computing resource budget, fewer backups can be deployed, and the less expensive computing resource can be utilized prior to the expensive computing resource by algorithm `Alg.2`.

8 Conclusion

In this paper, we investigated the user satisfaction problem in an MEC network through enhancing service reliability. We first formulated a novel optimization problem and showed its NP-hardness. We then proposed a constant approximation algorithm for the problem without the computing resource budget constraint. We also devised an approximation algorithm for the problem under a given computing resource budget constraint, the approximate solution is achieved at the expense of a moderate budget violation. We finally evaluated the performance of the proposed algorithms through simulations. Simulation results demonstrated that the proposed algorithms outperform the comparison baseline algorithms, improving the performance by more than 16.1% in comparison with the baseline algorithms.

Acknowledgments

The authors appreciate the three 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] L. Chen and J. Xu, “Budget-constrained edge service provisioning with demand estimation via bandit learning,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2364–2376, Oct. 2019.
[3] R. Cohen, L. Katzir, and D. Raz, “An efficient approximation for the generalized assignment problem,” *Inf. Process. Lett.*, vol. 100, pp. 162–166, 2006.
[4] P. Dai, K. Hu, X. Wu, H. Xing, and Z. Yu, “Asynchronous deep reinforcement learning for data-driven task offloading in MEC-empowered vehicular networks,” in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.
[5] V. Eramo, E. Miucci, M. Ammar, and F. G. Lavacca, “An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures,” *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 2008–2025, Aug. 2017.
[6] J. Fan, C. Guan, Y. Zhao, and C. Qiao, “Availability-aware mapping of service function chains,” in *Proc. IEEE Conf. Comput. Commun.*, 2017, pp. 1–9.
[7] X. Fu, F. R. Yu, J. Wang, Q. Qi, and J. Liao, “Dynamic service function chain embedding for NFV-enabled IoT: A deep reinforcement learning approach,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 507–519, Jan. 2020.
[8] GT-ITM. 2019. [Online]. Available: http://www.cc.gatech.edu/projects/gtitm/

[9] N. Guizani and A. Ghafoor, "A network function virtualization system for detecting malware in large IoT based networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 6, pp. 1218–1228, Jun. 2020.

[10] H. Guo, W. Huang, J. Liu, and Y. Wang, "Inter-server collaborative federated learning for ultra-dense edge computing," *IEEE Trans. Wireless Commun.*, vol. 21, no. 7, pp. 5191–5203, Jul. 2022.

[11] H. Guo, J. Li, J. Liu, N. Tian, and N. Kato, "A survey on space-air-ground-sea integrated network security in 6G," *IEEE Commun. Surv. Tuts.*, vol. 24, no. 1, pp. 53–87, 1st Quarter 2022.

[12] J. Halpern and C. Pignataro, "Service function chaining (SFC) architecture," RFC 7665, Oct. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc7665.txt

[13] F. He, T. Sato, and E. Oki, "Optimization model for backup resource allocation in middleboxes with importance," *IEEE/ACM Trans. Netw.*, vol. 27, no. 4, pp. 1742–1755, Aug. 2019.

[14] 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.

[15] J. Jia, L. Yang, and J. Cao, "Reliability-aware dynamic service chain scheduling in 5G networks based on reinforcement learning," in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.

[16] P. Jin, X. Fei, Q. Zhang, F. Liu, and B. Li, "Latency-aware VNF chain deployment with efficient resource reuse at network edge," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 267–276.

[17] Y. Kanizo, O. Rottenstreich, I. Segall, and J. Yallouz, "Optimizing virtual backup allocation for middleboxes," *IEEE/ACM Trans. Netw.*, vol. 25, no. 5, pp. 2759–2772, Oct. 2017.

[18] E. L. Lawler, "Fast approximation algorithms for knapsack problems," *Math. Oper. Res.*, vol. 4, no. 4, pp. 339–356, 1979.

[19] 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.

[20] 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.

[21] 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.

[22] J. Li, W. Liang, Z. Xu, X. Jia, and W. Zhou, "Service provisioning for multi-source IoT applications in mobile edge computing," *ACM Trans. Sensor Netw.*, vol. 18, no. 2, May 2022, Art. no. 17.

[23] 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.

[24] 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. Serv. Comput.*, early access, May 24, 2022, doi: 10.1109/TSC.2022.3176576.

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

[26] 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.

[27] 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.*, early access, May 18, 2021, doi: 10.1109/TMC.2021.3081681.

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

[29] J. Liu, H. Guo, J. Xiong, N. Kato, J. Zhang, and Y. Zhang, "Smart and resilient EV charging in SDN-enhanced vehicular edge computing networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 1, pp. 217–228, Jan. 2020.

[30] Y. Ma, W. Liang, and J. Wu, "Online NFV-enabled multicasting in mobile edge cloud networks," in *Proc 39th IEEE Int. Conf. Distrib. Comput. Syst.*, 2019, pp. 821–830.

[31] Y. Ma, W. Liang, J. Wu, and Z. Xu, "Throughput maximization of NFV-enabled multicasting in mobile edge cloud networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 2, pp. 394–407, Feb. 2020.

[32] E. Moro and I. Filippini, "Joint management of compute and radio resources in mobile edge computing: A market equilibrium approach," *IEEE Trans. Mobile Comput.*, early access, Jun. 23, 2021, doi: 10.1109/TMC.2021.3091764.

[33] B. Németh, N. Molner, J. Martín-pérez, C. J. Bernardos, A. De la Oliva, and B. Sonkoly, "Delay and reliability-constrained VNF placement on mobile and volatile 5G infrastructure," *IEEE Trans. Mobile Comput.*, vol. 21, no. 9, pp. 3150–3162, Sep. 2022.

[34] D. Pisinger, *Algorithms for Knapsack Problems*. Princeton, NJ, USA: Citeseer, 1995.

[35] L. Qu, M. Khabbaz, and C. Assi, "Reliability-aware service chaining in carrier-grade softwarized networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 558–573, Mar. 2018.

[36] 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, doi: 10.1109/TMC.2020.3048885.

[37] T. Taleb, A. Ksentini, and B. Sericola, "On service resilience in cloud-native 5G mobile systems," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 483–496, Mar. 2016.

[38] P. K. Thiruvasagam, A. Chakraborty, A. Mathew, and C. S. R. Murthy, "Reliable placement of service function chains and virtual monitoring functions with minimal cost in softwarized 5G networks," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 2, pp. 1491–1507, Jun. 2021.

[39] Y. Wang et al., "Reliability-oriented and resource-efficient service function chain construction and backup," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 1, pp. 240–257, Mar. 2021.

[40] 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.

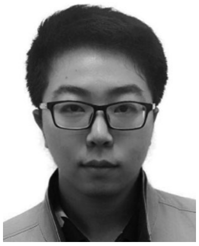
[41] Z. Xu et al., "Near optimal learning-driven mechanisms for stable NFV markets in multi-tier cloud networks," *IEEE/ACM Trans. Netw.*, early access, Jun. 08, 2022, doi: 10.1109/TNET.2022.3179295.

[42] L. Yala, P. A. Frangoudis, and A. Ksentini, "Latency and availability driven VNF placement in a MEC-NFV environment," in *Proc. IEEE Glob. Commun. Conf.*, 2018, pp. 1–7.

[43] S. Yang, F. Li, S. Trajanovski, R. Yahyapour, and X. Fu, "Recent advances of resource allocation in network function virtualization," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 2, pp. 295–314, Feb. 2021.

[44] 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.

[45] Q. Zhang, F. Liu, and C. Zeng, "Online adaptive interference-aware VNF deployment and migration for 5G network slice," *IEEE/ACM Trans. Netw.*, vol. 29, no. 5, pp. 2115–2128, Oct. 2021.

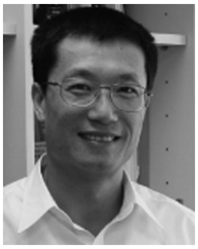


Jing Li received the BSc degree with the first class Honours in computer science from Australian National University, in 2018. He currently working toward the PhD degree with the School of Computing, Australian National University. His research interests include mobile edge computing, network function virtualization, and combinatorial optimization.

Weifa Liang (Senior Member, IEEE) received the BSc degree from Wuhan University, China, in 1984, the ME degree from the University of Science and Technology of China, in 1989, and the PhD degree from Australian National University, in 1998, all in computer science. He is a professor with the Department of Computer Science, City University of Hong Kong. Prior to that, he was a 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), Digital Twins and Internet of Things, Software-Defined Networking (SDN), design and analysis of parallel and distributed algorithms, approximation algorithms, combinatorial optimization, and graph theory. He currently serves as an editor in the editorial board of *IEEE Transactions on Communications*.

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 with 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.

Zichuan Xu (Member, IEEE) received the BSc and ME degrees from the Dalian University of Technology, China, in 2008 and 2011, respectively, all in computer science, and the PhD degree from The Australian National University, Australia, 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 professor and a PhD advisor with the School of Software, Dalian University of Technology, China. He is also a 'Xinghai Scholar' with the Dalian University of Technology, China. His research interests include cloud computing, mobile edge computing, deep learning, network function virtualization, software-defined networking, routing protocol design for wireless networks, algorithmic game theory, and optimization problems.



Xiaohua Jia (Fellow, IEEE) received the BSc and ME degrees from the University of Science and Technology of China, in 1984 and 1987, respectively, and DSc degree in information science from the University of Tokyo, in 1991. He is currently a chair professor with the Department of Computer Science, City University of Hong Kong. His research interests include cloud computing and distributed systems, computer networks, wireless sensor networks and mobile wireless networks. He is an editor of *IEEE Transactions on Parallel and Distributed Systems* (2006-2009), *Journal of World Wide Web*, *Wireless Networks*, *Journal of Combinatorial Optimization*, and so on. He is the general chair of ACM MobiHoc 2008, TPC co-chair of IEEE MASS 2009, area-chair of IEEE INFOCOM 2010, TPC co-chair of IEEE GlobeCom 2010, Ad Hoc and Sensor Networking Symposium, and Panel co-chair of IEEE INFOCOM 2011.

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 with the University of Sydney. To date, he has published more than 700 scientific papers and articles and is (co-)author/editor of more than 30 books. A sought-after speaker, he has delivered more than 250 keynote addresses, invited seminars, and media briefings. He is currently the editor in chief of the *ACM Computing Surveys* and served in the past as editor in chief of the *IEEE Transactions on Computers* (2010-2014) and the *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. 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 he recent awards include the New South Wales Premier's Prize of Excellence in Engineering and Information and Communications Technology (2019) and the Research Innovation Award, IEEE Technical Committee on Cloud Computing (2021). His research interests lie in parallel and distributed computing, networking, and complex systems.

Song Guo (Fellow, IEEE) is a full professor with the Department of Computing, The Hong Kong Polytechnic University. He also holds a changjiang chair professorship awarded by the Ministry of Education of China. He is a fellow of the Canadian Academy of Engineering and a fellow of the IEEE (Computer Society). His research interests are mainly in Big Data, edge AI, mobile computing, and distributed systems. He published many papers in top venues with wide impact in these areas and was recognized as a Highly Cited Researcher (Clarivate Web of Science). He is the recipient of over a dozen best paper awards from IEEE/ACM conferences, journals, and technical committees. He is the editor-in-chief of *IEEE Open Journal of the Computer Society* and the chair of IEEE Communications Society (ComSoc) Space and Satellite Communications Technical Committee. He was an IEEE ComSoc distinguished lecturer and a member of IEEE ComSoc Board of Governors. He has served for IEEE Computer Society on Fellow Evaluation Committee, and been named on editorial board of a number of prestigious international journals like *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Cloud Computing*, *IEEE Transactions on Emerging Topics in Computing*, etc. He has also served as chairs of organizing and technical committees of many international conferences.