Online Reliability-Enhanced Virtual Network Services Provisioning in Fault-Prone Mobile Edge Cloud

Yu Qiu, Junbin Liang, Victor C. M. Leung, *Life Fellow, IEEE*, Xu Wu, and Xia Deng, *Member, IEEE*

***Abstract*— Fault-Prone Mobile Edge Cloud (FP-MEC) is a new type of distributed network composed of mobile edge computing and network function virtualization, where virtual network services can be provided in the form of service function chains (SFCs) that are a sequence of virtual network functions (VNFs) on-demand deployed on resource-limited edge servers. FP-MEC has a characteristic that the fault probability of each VNF is dynamic and fluctuates with time and workloads, making SFCs temporarily unreliable. To increase the reliabilities, redundant Backup VNFs (BVNFs) need to be deployed near the VNFs and activated when they experience faults. Different mobile users would request different SFCs with reliability and service time demands to process their data. However, workloads of VNFs are dynamic and unpredictable in FP-MEC due to random arrival of user requests. How to optimally deploy VNFs and corresponding BVNFs on a set of edge servers to form expected SFCs that have higher reliabilities than user demand values, meanwhile throughput of receiving requests is maximized while receiving cost is minimized in real-time, is a challenging problem. The receiving cost is composed of deployment cost of instantiating VNFs and BVNFs, and communication cost of routing data among users, VNFs and BVNFs. In this paper, the long-term provisioning problem is first formulated as an integer linear program and proved to be NP-hard. Then, it is discretized into a sequence of one-slot optimization problems to handle practical time-varying fault probability, where a set of SFC requests are given at each time slot, and receiving or rejecting decisions are executed immediately without any future information. Finally, an online approximation scheme with a constant approximation ratio is proposed to solve the one-slot problems in polynomial time. Theoretical analyses and experiments based on real network topology of CERNET in China demonstrate that the scheme is promising compared to existing works.**

***Index Terms*— Mobile edge cloud (MEC), network function virtualization (NFV), service function chain (SFC), reliability, fault, online.**

Manuscript received 8 July 2021; revised 16 November 2021 and 16 January 2022; accepted 4 March 2022. Date of publication 15 March 2022; date of current version 12 September 2022. This work was supported in part by the Natural Science Foundation of Guangxi Province under Grant 2019GXNSFAA185042, in part by the Guangxi Key Research and Development Plan Project under Grant Guike AB19259006, in part by the National Natural Science Foundation of China under Grant 62062006 and Grant 61702127, in part by the Shenzhen Science and Technology Innovation Commission under Grant R2020A045, in part by the Science and Technology Program of Guangzhou under Grant 201804010461, in part by the China Scholarship Council under Grant 201908440064, and in part by the Natural Science Basis Research Plan in Guangxi Province of China under Program 2018GXNSFAA281292. The associate editor coordinating the review of this article and approving it for publication was X. Gong. *(Corresponding author: Junbin Liang.)*

Yu Qiu and Junbin Liang are with the Guangxi Key Laboratory of Multimedia Communications and Network Technology, School of Computer and Electronics Information, Guangxi University, Nanning 530004, China (e-mail: qyu_edu@163.com; liangjb@gxu.edu.cn).

Victor C. M. Leung is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China.

Xu Wu is with the School of Computer and Electronics Information, Guangxi University, Nanning 530004, China.

Xia Deng is with the School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou 510006, Guangdong, China.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TWC.2022.3157606.

Digital Object Identifier 10.1109/TWC.2022.3157606

I. Introduction

The Fault-Prone Mobile Edge Cloud (FP-MEC) consists of access points, edge servers, and mobile user devices, combining mobile edge computing and network function virtualization to provide various virtual network services (e.g., virtual reality and face recognition) on resource-limited edge servers. FP-MEC shortens logical and physical distances between users and edge servers [1], enabling user devices to obtain tailor-made service function chains (SFCs) composed of different interconnected virtual network functions (VNFs) in low latency and low energy consumption way. Besides the specific connection sequence of VNFs [2], different users also have different requirements on SFC reliability and service time. The reliability is the ability of a network to maintain stable SFC services to achieve a reliable value higher than users expected levels in facing potential faults [3].

Deploying redundant Backup VNFs (BVNFs) near VNFs is a practical way to increase the reliability. That is, one or multiple BVNFs are added near a single VNF to satisfy the minimum reliability threshold of each user [4]. A feature of FP-MEC that VNFs fault probabilities and reliabilities are not static but vary over time and workloads is explicitly observed in the article, as shown in Fig. 1[1] and Fig 2. The feature[2] causes the previously provided BVNFs to be insufficient for

[1]The Fig. 1 shows an experiment exploring relationship between VNF fault probability and its workload by varying number of arrival requests from 100 to 1000 during 10 min. Virtual Nginx is a VNF instance with load balancing function. Alibaba is an edge cloud platform provider in China.

[2]Note that fluctuated VNF reliabilities have little impact on SFC reliabilities in data center environments that are different from the FP-MEC environment studied in this article. This is because data centers are always rich in computing and storage resources and can deploy enough BVNFs to achieve high reliability levels (e.g., 99.999%) for mobile users.

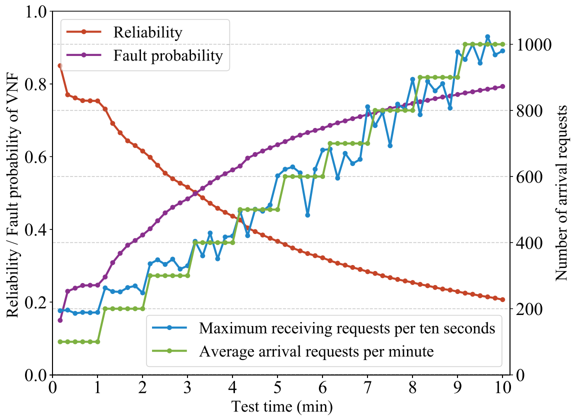


Fig. 1. Experimental results based on virtual Nginx on Alibaba.

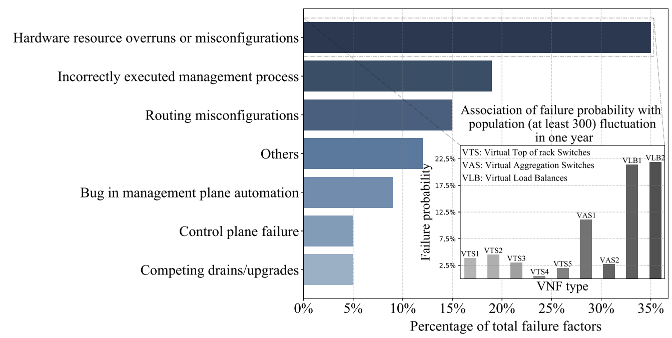


Fig. 2. Factors of causing server failure in Google, and association between population and fault [5], [6].

ensuring reliabilities in real-time. Specifically, random arrival SFC requests could share some common VNFs to reduce their deployment costs [7], but result in heavier and more dynamic workloads for the VNFs during sharing. VNFs are more prone to various software errors or hardware malfunctions under high workloads [5], [6], [8], [9]. Therefore, the fault probabilities of the VNFs will rise, which causes the SFCs temporarily unreliable due to BVNFs insufficiency.

However, arbitrary addition and deployment of BVNFs for each VNF in SFCs not only rapidly exceeds capacities of resource-limited edge servers, but also increases data routing distances from a VNF to these locations where corresponding BVNFs are placed [10]. In addition, dynamic and unpredictable workloads of VNFs would cause SFC reliabilities to fluctuate over time. Considering these phenomena in resource-limited FP-MEC, how to implement online reliability-enhanced SFCs provisioning poses several significant challenges: 1) how to optimally select the deployment locations of VNFs to meet service sequence and time requirements, 2) how to optimally select which VNFs in each SFC should add redundant BVNFs, 3) where and how much BVNFs need to be placed optimally to meet reliability requirements.

In this paper, the long-term provisioning problem is first formulated as an integer linear program, and proved to be NP-hard by reducing it to the Lowest cost Generalized Assignment Problem (LGAP) [11], [12]. In addition, two cost measures are considered jointly to make receiving cost model more accurate: communication cost of routing data among users and locations where requested VNFs and BVNFs are deployed, and deployment cost of instantiating these functions. To the best of our knowledge, we are the first to explicitly insight the fault-prone nature of VNFs, and jointly consider the sharing of VNFs and BVNFs in SFCs provisioning. Then, the problem is discretized into a series of one-slot optimization problems to handle practical time-varying fault probability, where a set of SFC requests are given at each slot, and receiving or rejecting decisions are performed immediately without any future information.

Finally, we propose a two-stage online scheme to solve the one-slot problems by striving to trade-off among limited capacity, the sharing of VNFs and BVNFs, and fluctuated reliability to maximize throughput of receiving requests while minimizing receiving cost. In the first stage, the scheme selects where to deploy different VNFs to form SFCs for different service requests to minimize VNFs deployment and communication cost. However, SFC reliabilities may not always be higher than user expectations because the VNFs are fault-prone under high workloads or are less reliable due to their own properties (e.g., type or location). Therefore, the second stage adaptively decides which VNFs need to be protected, and numbers and locations of BVNFs to enhance SFC reliabilities while minimizing BVNFs deployment and communication cost.

The main contributions of this paper are as follows:

- To the best of our knowledge, we are the first to formulate the online reliability-enhanced SFCs provisioning problem in FP-MEC as integer linear program from a new perspective, which considers not only fault-prone nature of VNFs, but also the sharing of VNFs and BVNFs.
- To capture and handle time-varying fault probabilities of VNFs, the long-term provisioning problem is discretized into a sequence of one-slot optimization problems. Both kinds of issues are proved to be NP-hard.
- To reduce the hardness, we propose a two-stage online scheme with a constant approximation ratio. The scheme can solve the one-slot problems with deployment location, reliability, service time, service order, and capacity constraints in polynomial time.
- To evaluate effectiveness of the scheme, we conduct extensive experiments based on real network topology of CERNET in China. The results demonstrate that the scheme could reduce receiving cost by at least 10% and effectively adapt to fault-prone networks compared to state-of-the-art works.

The rest of this article is organized as follows. Section II reviews relevant works. Section III introduces related mathematical models and symbols, and proves the problem to be NP-hard. In Section IV, we give an overview of the two-stage online scheme and describe in detail an algorithm and mathematical analysis of the first stage. Section V introduces related algorithm of the second stage. Section VI evaluates scheme performances. Section VII summarizes full text.

II. Related Work

Recently, reliable VNFs/SFCs provisioning problem has attracted the attention of many scholars. In overviews [8], [13], [14], problems and solutions of fault type, detection, and recovery in mobile edge cloud are elaborated. According to the principle of whether SFCs share VNFs or BVNFs, above researches are divided into three categories: dedicated VNF and dedicated BVNF, dedicated VNF and shared BVNF, and shared VNF and dedicated BVNF.

A. Dedicated VNF and Dedicated BVNF

In 2017, Qu *et al.* [15] modeled SFCs provisioning problem as a mixed integer linear program. He adopted a heuristic algorithm without considering node processing delay to ensure reliability and routing delay constraints by trading-off between single routing path and multi routing path. In 2018, Fan *et al.* [16] assumed that servers with static fault probability ran in an normal alternating with fault environment, and used an approximation algorithm to reduce BVNFs deployment cost. In 2020, Li *et al.* [17] considered a variant (online model) of [3], whose goal was to maximize numbers of SFC requests with reliability requirements over a period of time. To improve deployed SFC reliabilities, Liang *et al.* [4] deployed BVNFs on edge servers with sufficient resources in the k-hop range of servers hosting VNFs. When fault probabilities were unknown, Shang *et al.* [18] used a scheme combining static and dynamic backup methods to provide reliable services in edge-cloud networks. Yang *et al.* [19] deployed K backup SFCs to substrate networks through a heuristic algorithm. Even when $K-1$ backup SFCs fault, those primary SFCs still provide continuous services.

B. Dedicated VNF and Shared BVNF

In 2015, Fan *et al.* [20] was one of the first to study shared BVNFs. They proposed a joint protection scheme, which simultaneously selected the two least reliable VNFs and provided shared BVNFs to protect them, improving iteratively a single SFC reliability. In 2017, Fan *et al.* [21] used a heuristic algorithm to deploy a single reliable SFC to data centers in both on-site and off-site scenarios, reducing BVNFs numbers. In addition, a shared BVNFs pool mechanism was introduced when deployed multiple SFCs. Kanizo *et al.* [22] adopted an offline shared BVNFs scheme to improve full survival probability of fault nodes. In addition, they also considered resisting small-range faults through pre-deploying BVNFs when resource was limited. However, they only considered limited number of backups and ignored dependencies between VNFs. Qu *et al.* [23] adopted a centralized architecture that combined SFCs routing and reliability to meet user reliability requirements by providing shared BVNFs for adjacent VNFs in a SFC. In 2019, Zhang *et al.* [24] considered various VNF resource demands, and allocated resource to shared BVNFs through a joint reliability and resource utilization algorithm.

C. Shared VNF and Dedicated BVNF

In 2017, Ding *et al.* [25] proposed a heuristic algorithm based on Birnbaum Importance Measure (BIM). The algorithm integrated important VNFs (e.g. a VNF shared by multiple SFCs) on a highly reliable server and provided backups for them. In 2019, Dinh and Kim [26] adopted a backup resource allocation algorithm based on Criticality Importance Measure (CIM) to improve [25]. Improved algorithm jointly considered resource consumption of shared VNFs and their contribution to global reliability. Li *et al.* [7] aimed to receive as many services as possible in a single cloud by balancing shared SFC and its requested backup resources.

However, above articles [3], [4], [7], [15]–[17], [19]–[27] assume that **VNF fault probabilities are static**. Therefore, in a more dynamic scenario, they are difficult to guarantee reliability demands and provide optimal solutions in real-time.

Novelty: The primary novel aspect is the problem we consider in this paper, i.e., online reliability-enhanced SFCs provisioning in resource-limited FP-MEC, which proved to be NP-hard. Specifically, distinctions with existing efforts are as follows: 1) dynamic VNF fault probability that fluctuates with its workload, 2) dynamic network topology that changes due to the arrival and departure of users with different service time requirements, 3) the sharing of VNFs and BVNFs are introduced concurrently to reduce instantiating number of them and improve resource utilization of edge servers.

III. Problem Description

In this section, we first present associated system models, notations, and formulas for the online reliability-enhanced SFCs provisioning in resource-limited FP-MEC. Then, the problem is accurately defined and discretized as a sequence of one-slot optimization problems. Finally, both kinds of problems are proved to be NP-hard.

A. Substrate Network

We consider a resource-limited FP-MEC to be an undirected weighted graph $G^s = (V^s, E^s)$. V^s consisting of C and V represents all physical nodes in the G^s. Specifically, C is a group of service nodes $c \in C$ connecting to edge servers with computing capacity $cap_c > 0$. V is a set of transmission nodes (access points) $v \in V$, which only transmits data and has no computing capacity $cap_v = 0$. E^s represents a set of links $e = (u, v)$ $(u, v \in V^s)$ to connect any two peers in G^s. Each link e has bandwidth B_e and distance D_e. For the sake of convenience, we assume that each e is a high-capacity optical cable, which refers to any e has same and sufficient bandwidth capacity (e.g., $B_e = 1$). The assumption has little influence on accuracy of the proposed scheme, as shown in Fig 13. In addition, the assumption has been adopted in literatures [10], [28], which could be extended to scenarios with different and limited bandwidth capacity.

B. User Requests

Φ and Γ represent all types of VNFs and a diverse set of SFC request γ $(S^\gamma, T_a^\gamma, T_f^\gamma, R^\gamma, F^\gamma)$, respectively.

TABLE I
RELEVANT SYMBOL

Symbol	Description	Symbol	Description
Topology related:		**Request related**:	
C	A set of edge server c with $cap_c > 0$	$A_{remain}(t)$	A set of receiving requests that does not meet service time demand before t slot
V	A group of accept point with $cap_v = 0$	$A_{rrive}(t)$	A set of new arrival requests at t slot
V^s	All physical nodes, $C \cup V$	$A_{ccept}(t)$	A set of receiving requests at t slot
$e = (u, v)$, $P_{u,v}$	A link connecting any two nodes in V^s. A set of links contained in the shortest physical path from u to v	$\gamma(S^\gamma, T_a^\gamma, T_f^\gamma, R^\gamma, F^\gamma)$	S^γ is user location, T_a^γ and T_f^γ are arrival and departure slot, R^γ is reliability demands, and F^γ is a group of VNFs
Reliability related:		**Resource related**:	
$\rho'_f(\rho_f)$	The initial (actual) fault probability of VNF f	cap_f	The computing demand of VNF f
$S_c^i(S_c^{ib})$, δ	The number of sharing $f_i(f_{ib})$ in edge server c, the workload sensitivity	cap_c	The total computing capacity of server c
$r^i(\hat{r}^i)$	The actual reliability of unprotected (protected) f_i	$cap_c(t)$	The remaining computing capacity of server c at t slot
$\mathcal{R}^\gamma(\hat{\mathcal{R}}^\gamma)$	The reliability of unprotected (protected) γ	D_e	The distance of link e
Cost related:		**Variable related**:	
C_{cu},C_{du},C_{bu}	The cost per unit of computing, distance, bandwidth	X^γ	Whether all VNFs in request γ is deployed
C_{f_i}	The deployment cost of VNF f_i	$x_{c_j}^{\gamma,i}$	Whether f_i in γ is deployed on c_j
C_γ^{dep}	The deployment cost of request γ	$y_{c_j}^{\gamma,ib_k}$	Whether the k-th backup of f_i in γ is deployed on c_j
C_γ^{tra}	The transmission cost among VNFs in request γ and user	$x_{u,v}^{l_b(f_i,f_{ib_k})}$	Whether the f_i and f_{ib_k} are deployed on u, v in γ
$C_{\gamma b}^{dep}$	The BVNFs deployment cost of γ	$z_{u,v}^{l(f_i,f_{i+1})}$	Whether the f_i and f_{i+1} are deployed to u, v in γ
C_γ^{tsyn}	The synchronization cost among VNFs and its BVNFs in γ	$s_{c_j}^i(s_{c_j}^{ib_k})$	Whether to share VNF (BVNFs) f_i in c_j

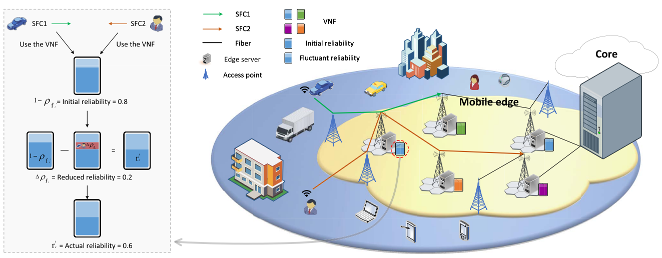


Fig. 3. FP-MEC network.

In particular, S^γ is location of a mobile user device, T_a^γ and T_f^γ is defined as arrival and departure time slot of request γ, respectively. R^γ represents a reliability value of user expected. F^γ is a group of VNFs f_i $(0 < i \leq |F^\gamma|)$ with capacity demands cap_{f_i} and service order demands, which are sequentially connected by several virtual links $L^\gamma = \cup(f_i^\gamma, f_{i+1}^\gamma)$. A specific SFC provisioning example is shown in Fig. 3. A mobile device first sends a SFC request γ (e.g., SFC1 or SFC2) to the FP-MEC control center. Then, all VNFs that form the SFC are deployed on a series of service nodes, consuming computing resource $\sum\limits_{i \in F^\gamma} cap_{f_i}$. Finally, the control center automatically implements data routing between adjacent VNFs by mapping virtual links L^γ to the shortest path consisted a set of physical links P^γ. Noticing, when a user in an overlapping area covered by multiple access points, we assume that the user device will connect to the closest access point or the one with the strongest signal strength [10].

C. Fault and Reliability Model

In resource-limited FP-MEC, reliability is a major factor affecting processes of SFCs provisioning. In the fault and reliability model, running time of a VNF is divided into uptime and repair (fault) time. Reliability r_f of VNF f is expressed as not only a probability that is a fraction of uptime to total time, but also a opposite event with fault probability ρ_f in many literatures included in the review [29]. Reliability r_f and fault probability ρ_f are defined as follows:

$$r_f = \frac{T_{normal}}{T_{all}} = 1 - \rho_f \tag{1}$$

$$\rho_f = \frac{T_{repair}}{T_{all}} = \frac{MTTR}{MTTN + MTTR} \tag{2}$$

MTTN and MTTR are Mean Time To Normal and Mean Time To Repair, respectively.

Theorem 1: In resource-limited FP-MEC, the increment of fault probability of a shared VNF is positively correlated with the request number of sharing it.

Proof: Mathematical induction and contradiction are used to prove the theorem as follows. $\mathbb{S}$ represents the request number of sharing VNF f. Firstly, when $\mathbb{S} = 0$, f is not used or shared. ρ_f is determined by initial fault probability ρ'_f of f based on its own properties (e.g., type or location). Secondly, when $\mathbb{S} \geq 1$, ρ_f increases with $\mathbb{S}$. Finally, when $\mathbb{S} = \eta$, **assume that ρ_f is independent of** η. Two states of edge servers need to be considered: capacity sufficient and insufficient, where η' represents the critical threshold of sharing number between two states.

Case 1: Remaining capacities are sufficient indicates that new arrival requests can be accepted $(\mathbb{S} = \eta < \eta')$, as shown by blue line in the Fig. 1, where the maximum receiving request numbers is approximated as $\mathbb{S}$. Therefore, the Pearson product-moment correlation coefficient [30] $P_{\rho\mathbb{S}} \in [-1, 1]$ between ρ_f and $\mathbb{S}$ is:

$$P_{\rho\mathbb{S}} = \frac{\sum\limits_{t=1/6}^{10} (\rho_t - \bar{\rho})(\mathbb{S}_t - \bar{\mathbb{S}})}{\sqrt{\sum\limits_{t=1/6}^{10} (\rho_t - \bar{\rho})^2} \cdot \sqrt{\sum\limits_{t=1/6}^{10} (\mathbb{S}_t - \bar{\mathbb{S}})^2}} = 0.93$$

$$t \in \{1/6, 2/6, \ldots, 10\}$$

where, $\rho_t(\mathbb{S}_t)$ represents VNF fault probabilities (sharing number) at t time in Fig. 1, respectively. $\bar{\rho}(\bar{\mathbb{S}})$ is the mean value of all $\rho_t(\mathbb{S}_t)$. $P_{\rho\mathbb{S}} > 0$ **indicates there is a positive correlation, and the closer $P_{\rho\mathbb{S}}$ is to 1, the stronger positive correlation is**. In addition, [31] and [32] also prove that fault probabilities of hardware or software environments running VNFs, including edge servers and virtual machines, are positively correlated with their workload.

Case 2: When the remaining resources are insufficient, consider an extreme case. That is, when $\lim_{\eta \to +\infty} \mathbb{S} = +\infty > \eta'$, f cannot handle η requests at the same time due to limited memory or CPU resources in real service scenarios (e.g., Fig 2 and [5], [6], [8], [9]), which incurs fault.

So, **the assumption is wrong**. In addition, $\rho_f = \Delta(\rho_f) + \rho'_f$, where initial fault probability ρ'_f is a constant, and $\Delta(\rho_f)$ represents the increment of fault probability of f. Therefore, $\Delta(\rho_f)$ is positively correlated with $\mathbb{S}$. That is the proof. □

Corollary 1: If the number of sharing f_i in any server c is different, then f_i would have different fault probability ρ_{f_i} in each c.

Proof: According to formula (2), ρ_{f_i} is proportional to MTTR (fault time), when total running time is fixed. Therefore, we only need to prove that fault time is proportional to the sharing number in server c. The reasons as follows: 1) when a new arrival user shares VNF, the configuration operations need to be performed again. The frequency of VNF reconfiguration increases as the sharing number rises, which leads to an increase in faults caused by reconfiguration errors [33]. So total fault time would be longer. The more important factor is 2) the larger sharing number causes VNFs to suffer from heavier workloads. The VNFs are more prone to various software errors or hardware malfunctions under high workloads [5], [8], which is proved in Fig. 1 and Theorem 1. This also increases total fault time. That is the proof. □

To describe accurately fluctuation amplitude of fault probability, we let $\mathbb{S} = \bigcup_{i \in \Phi, c \in C} S_c^i$ represents a set of numbers of sharing f_i in any service node c. Let δ is a proportionality coefficient between the increment of ρ_{f_i} and the number of sharing f_i. Note that δ represents workload sensitivity of VNFs. As δ increases, VNFs are more prone to fail under high workload, which reduces the amount of sharing VNFs in any edge servers. Therefore, some VNFs have to be instanced and deployed on servers far from users due to tight resources, increasing communication cost. The fluctuated fault probability $\Delta(\rho_{f_i})$ and actual reliability r_c^i of f_i deployed on c is represented as:

$$\Delta(\rho_{f_i}) = \delta \cdot S_c^i, \quad r_c^i = 1 - \left[\Delta(\rho_{f_i}) + \rho'_{f_i}\right] \tag{3}$$

Blue VNF in Fig. 3 is an example of formula (3), where $\delta = 0.1$ and $S_c^i = 2$. We assume that each VNF fault occurs independently and does not affect and extend to other VNFs. Therefore, the reliability $\mathcal{R}^\gamma$ of unprotected SFC is product of reliabilities of its included VNFs:

$$\mathcal{R}^\gamma = \prod_{f_c^i \in F^\gamma} r_c^i \tag{4}$$

However, unprotected $\mathcal{R}^\gamma$ is always lower than reliability requirement R^γ of user, because VNFs are fault-prone under high workloads or are less reliable due to their own properties. Therefore, we provide shared BVNFs for same type of VNFs that shared by multiple SFCs for ensuring continuous SFC services. In real service scenarios, probability of the same type of VNFs simultaneous fault in multiple locations is extremely small. Based on this phenomenon, we assume that only a VNF of each type experiences fault at any time, but different types of VNFs could experience faults simultaneously. In addition, a protected VNF f_i (f_{ib_0}) faults only if its a set of BVNFs $F_{ib} = \{f_{ib_1}, f_{ib_2}, \ldots, f_{ib_k}\}$ all fault at the same time. Therefore, the reliability $\hat{r}^i$ $(\hat{\mathcal{R}}^\gamma)$ of protected f_i (γ) is expressed as follows when f_i have k shared backups:

$$\begin{aligned} \hat{r}^i &= \begin{cases} 1 - \left(1 - r_c^i\right) \cdot \prod_{k \ge 1, j \ge 1}^{k \le |ib_k|, j \le |C|} \left(1 - r_{c_j}^{ib_k} y_{c_j}^{\gamma, ib_k}\right), & k \ge 1 \\ r_c^i, & k = 0 \end{cases}, \\ \hat{\mathcal{R}} &= \prod_{f_i \in F^\gamma} \hat{r}^i \end{aligned} \tag{5}$$

$y_{c_j}^{\gamma, ib_k}$ is the binary variable, and $y_{c_j}^{\gamma, ib_k} = 1$ indicates that the k-th backup of f_i in γ is deployed on server c_j, otherwise 0. To analyze influences of all BVNF reliabilities on performances and speed up algorithm, we use expectation value $\mathbb{E}[r^b]$ of reliabilities of all backups to approximately represent each BVNF reliability. The expectation value has little influence on algorithm performances and could change based on actual scenarios as shown in Fig. 8 and Fig. 14.

D. Receiving Cost Model

In resource-limited FP-MEC, receiving cost is another major factor affecting SFCs provisioning. In the receiving cost model, we mainly consider deployment cost C^{dep} and communication cost C^{com}.

1) Deployment Cost: The C^{dep} is consisted of VNF deployment cost C^{depi} and corresponding BVNF deployment cost C^{depib}. C_{cu} is cost of per unit computing capacity. Since same type VNFs and BVNFs require same computing capacity cap_{f_i}, the deployment cost C_{f_i} of f_i is:

$$C_{f_i} = C^{depi} = C^{depib} = C_{cu} \cdot cap_{f_i} \tag{6}$$

In sharing scenarios where VNFs are shared among multiple SFCs and BVNFs shared are among multiple VNFs, a request γ sharing existing VNFs and BVNFs requires additional communication cost, but does not require deployment cost. Therefore, VNFs deployment cost C_γ^{dep} and its BVNFs deployment cost $C_{\gamma b}^{dep}$ of γ are:

$$\begin{aligned} C_\gamma^{dep} &= \sum_{i \in F^\gamma, c_j \in C} C_{f_i} \cdot x_{c_j}^{\gamma, i} \cdot \left(1 - s_{c_j}^i\right), \\ C_{\gamma b}^{dep} &= \sum_{i \in F^\gamma, c_j \in C, ib_k \in F_{ib}^\gamma} C_{f_i} \cdot y_{c_j}^{\gamma, ib_k} \cdot \left(1 - s_{c_j}^{ib_k}\right) \end{aligned} \tag{7}$$

$x_{c_j}^{\gamma, i}$ is a binary variable that indicates whether f_i in γ is deployed on c_j. $s_{c_j}^i, s_{c_j}^{ib_k}$ are also a binary variable,

$s^i_{c_j}(s^{ib_k}_{c_j}) = 1$ means that $f_i(f_{ib})$ in c_j could be shared, otherwise, $s^i_{c_j}(s^{ib_k}_{c_j}) = 0$.

2) Communication Cost: C_{du} and C_{bu} are costs of per unit distance and bandwidth, respectively. The communication cost C^{com} is consisted of transmission cost C^{tra} among a user and requested VNFs, and synchronization cost C^{tsyn} among VNFs and corresponding BVNFs. Specifically, the transmission cost C^{tra}_{γ} is sum of C^{tran}_l ($l \in L^{\gamma}$). C^{tran}_l of f_i is cost of transmitting data using a virtual link $l(f_i, f_{i+1})$, which corresponds to the shortest physical path $P_{u,v}$ from u to v ($u, v \in V^s$).

$$\begin{aligned} C_{tu} &= C_{du}C_{bu}, \\ C^{tran}_l &= \sum_{e \in P_{u,v}} D_e B_e C_{tu}, \\ C^{tra}_{\gamma} &= \sum_{l \in L^{\gamma}} C^{tran}_l \cdot z^{l(f_i,f_{i+1})}_{u,v} \end{aligned} \tag{8}$$

The $z^{l(f_i,f_{i+1})}_{u,v}$ is a binary variable, and $z^{l(f_i,f_{i+1})}_{u,v} = 1$ means that VNFs f_i, f_{i+1} in γ are deployed in node u, v, respectively. otherwise $z^{l(f_i,f_{i+1})}_{u,v} = 0$.

In addition, frequent transmission of synchronization information among VNFs and corresponding BVNFs is necessary. When a VNF experiences faults, its redundant BVNFs can quickly and seamlessly take over traffics processed by the VNF through historical synchronization data. The synchronization cost C^{tsyn}_{γ} of request γ is sum of $C^{tsyn}_{l_b}$:

$$C^{tsyn}_{\gamma} = \sum_{l_b \in L^{\gamma}_b} C^{tsyn}_{l_b} \cdot x^{l_b(f_i,f_{ib_k})}_{u,v} \tag{9}$$

where, $x^{l_b(f_i,f_{ib_k})}_{u,v}$ is a binary variable, indicating whether f_i and its backup f_{ib_k} in γ are deployed in u, v. Therefore, the sum of VNFs deployment cost C^{dep}_{γ} and transmission cost C^{tra}_{γ}, and the sum of additional BVNFs deployment cost $C^{dep}_{\gamma b}$ and synchronization cost C^{tsyn}_{γ} for a set of requests are:

$$\begin{aligned} C_{DepS} &= \sum_{\gamma \in \Gamma} C^{dep}_{\gamma} + C^{tra}_{\gamma}, \\ C_{IncS} &= \sum_{\gamma \in \Gamma} C^{dep}_{\gamma b} + C^{tsyn}_{\gamma}, \\ C_{all} &= C_{DepS} + C_{IncS} \end{aligned} \tag{10}$$

E. Problem Definition

Without loss of generality, we first discrete a continuous time horizon $\mathbb{T}$ into a set of equally spaced time slots $\mathbb{T} = (t_1, t_2, \ldots, t_{|\mathbb{T}|})$. Each time slot represents a decision interval [34]. Therefore, the long-term optimal SFCs provisioning is discretized into a sequence of one-slot (offline) optimization problems, which is beneficial to capture and handle practical time-varying fault probability. Next, we define X^{γ} as a binary variable that indicates whether all VNFs contained in request γ are deployed. Function $[\cdot] = 1$ shows that condition $\cdot$ is true. Finally, we distinguish concepts among $A_{rrive}(t)$, $A_{ccept}(t)$, and $A_{remain}(t)$ as follows: the $A_{rrive}(t)$ is a set of new arrival requests that are given at each slot, the $A_{ccept}(t)$ is a set of receiving requests that requested VNFs and BVNFs are deployed on edge servers at t slot, and the $A_{remain}(t)$ is a group of receiving requests that does not meet service time requirements before t slot.

$$\begin{aligned} X^{\gamma} &= \min_{f_i \in \gamma}\Big(\sum_{c_j \in C} x^{\gamma,i}_{c_j}\Big), \\ A_{ccept}(t) &= \bigcup_{\gamma \in A_{rrive}(t)} \gamma \cdot X^{\gamma} \cdot \Big[\hat{\mathcal{R}}^{\gamma} \geq \mathrm{R}^{\gamma}\Big] \end{aligned} \tag{11}$$

The problem of Long-term (online) Reliability-enhanced SFCs Provisioning problem (LRSP) and its discrete One-slot (offline) problem (ORSP) in resource-limited FP-MEC are precisely defined as follows:

Definition 1: Given a resource-limited FP-MEC $G^s = (V^s, E^s)$ with a set C of edge servers with computing capacity $cap_c > 0$, a group of link e with D_e, and a set of requests $\gamma(S^{\gamma}, T^{\gamma}_a, T^{\gamma}_f, R^{\gamma}, F^{\gamma}) \in A_{rrive}(t)$ are given at begin of slot t, the ORSP problem is to receive as many requests as possible with minimal deployment cost of instantiating VNFs and BVNFs and communication cost between them such that maximizing the throughput of receiving requests $A_{ccept}(t)$ and minimizing receiving costs $C_{all}(t)$, while the reliability R^{γ}, service time T^{γ}_f and T^{γ}_a, service order F^{γ}, and capacity cap_{γ} demands of each received request γ are met, subject to resource capacity in G_s in one-slot.

Definition 2: The LRSP problem is to admit as many SFC requests as possible with minimal receiving cost without any future information such as arrival requests, server capacities, and VNF fault probabilities, while various demands of each request are met in G_s during time span $\mathbb{T}$, and receiving or rejecting decisions are performed at begin of each slot.

F. NP-Hardness of ORSP and LRSP Problems

Theorem 2: The ORSP and the LRSP problems in a FP-MEC $G^s = (V^s, E^s)$ are NP-hard.

Proof: We prove the NP-hardness of ORSP by a reduction from a well know NP-hard problem—**the Lowest cost Generalized Assignment Problem (LGAP)** that is defined as follows. Given a group of items $Item = (i_1, i_2, \ldots, i_n)$ with cost $cost_i$ and size $size_i$, and a set of bins $Bin = (b_1, b_2, \ldots, b_m)$ with capacity cap_{b_i}, the objective of LGAP is to pick as many items as possible into bins with minimal cost, while the total size of items in each bin does not exceed its capacity.

We demonstrate that an instance of the LGAP problem can be reduced to a specific case I of ORSP problem, where I is consisted of $G^s = (C \cup V, E^s)$ and a set of requests Γ. Specifically, each edge server $c \in C$ has computing capacity cap_c, and each request $\gamma \in \Gamma$ only contains one VNF with computing demand cap_f. All requests with the same service time arrive concurrently at begin of slot t. The cost C_{γ} of receiving a request is consisted of VNF deployment cost C^{depi}, and its required BVNF deployment cost C^{depib}, and synchronization cost $C^{tsyn}_{l_b}$ between them. To this end, the ORSP problem aims to receive as many requests as possible with minimal receiving cost, subject to capacities cap_c of edge servers.

Obviously, a solution to ORSP problem returns a solution to LGAP problem, and reduction an instance of the latter (LGAP) to an instance of the former takes polynomial time. As the LGAP problem is NP-hard, the ORSP problem is NP-hard, too. Assume that time span $\mathbb{T}$ contains only one time slot, e.g., $\mathbb{T} = 1$, the LRSP problem degenerates into the ORSP problem. Therefore, the LRSP problem is also NP-hard. □

The ORSP problem could be decomposed into following two subproblems, they are also NP-hard and its proof is like Theorem 2:

1) How to optimally select deployment locations of VNFs to meet service sequence and time demands of new arrival SFC requests $A_{rrive}(t)$.
2) How to optimally select which VNFs in each SFC should be added BVNFs, and how many and where to deploy to meet reliability demands of requests in $A_{rrive}(t)$ and $A_{remain}(t)$.

G. ILP Model

In this part, relevant symbols are first listed in the TABLE I. Then, LRSP problem is formulated as an integer linear program. Decision variables include $x_{c_j}^{\gamma,i}(t)$, $y_{c_j}^{\gamma,ib_k}(t)$, $z_{u,v}^{l(f_i,f_{i+1})}(t)$, and $x_{u,v}^{l_b(f_i,f_{ib_k})}(t)$. Long-term objective and restriction conditions of LRSP during time horizon $\mathbb{T}$ are:

Objective:

$$Minimize \sum_{t\in\mathbb{T}} C_{DepS}(t) + C_{IncS}(t) \tag{12}$$

$Subject\ to:$

$$C_{DepS}(t) = \sum_{\gamma\in A_{rrive}(t)} C_{\gamma}^{dep} + C_{\gamma}^{tra}, \quad t\in\mathbb{T} \tag{13}$$

$$C_{IncS}(t) = \sum_{\gamma\in A_{rrive}(t)\cup A_{remain}(t)} C_{\gamma b}^{dep} + C_{\gamma}^{tsyn}, \quad t\in\mathbb{T} \tag{14}$$

$$A_{ccept}(t) = \bigcup_{\gamma\in A_{rrive}(t)} \gamma\cdot X^{\gamma}(t)\left[\hat{\mathcal{R}}^{\gamma} \geq \mathrm{R}^{\gamma}\right], \quad t\in\mathbb{T} \tag{15}$$

$$\sum_{c_j\in C} x_{c_j}^{\gamma,i}(t) = 1, f_i\in\gamma, \gamma\in A_{ccept}(t),\ t\in\mathbb{T} \tag{16}$$

$$\sum_{c_j\in C} y_{c_j}^{\gamma,ib_k}(t) = 1, \quad ib_k\in F_{ib}^{\gamma}, \gamma\in A_{ccept}(t),\ t\in\mathbb{T} \tag{17}$$

$$S_c^i < \lfloor \frac{r_f}{\delta} \rfloor, \quad f_i\in\Phi, c\in C \tag{18}$$

$$\sum_{f_i\in\gamma} cap_{f_i}\cdot(x_{c_j}^{\gamma,i}(t)\cdot(1-s_{c_j}^i) + \sum_{ib_k\in F_{ib}^{\gamma}} y_{c_j}^{\gamma,ib_k}(t)\cdot(1-s_{c_j}^{ib_k}))$$
$$\leq cap_{c_j}(t), \quad c_j\in C, \gamma\in A_{ccept}(t)\cup A_{remain}(t),\ t\in\mathbb{T} \tag{19}$$

$$\prod_{f_i\in F^{\gamma}} \hat{r}^i \geq \mathrm{R}^{\gamma}, \quad \gamma\in A_{ccept}(t)\cup A_{remain}(t),\ t\in\mathbb{T} \tag{20}$$

$Variable:$

$$x_{c_j}^{\gamma,i}(t)\in\{0,1\},$$
$$c_j\in C, i\in\Phi, \gamma\in A_{rrive}(t),\ t\in\mathbb{T} \tag{21}$$

$$y_{c_j}^{\gamma,ib_k}(t)\in\{0,1\},$$
$$c_j\in C, ib_k\in F_{ib}^{\gamma}, \gamma\in A_{rrive}(t),\ t\in\mathbb{T} \tag{22}$$

$$z_{u,v}^{l(f_i,f_{i+1})}(t)\in\{0,1\},$$
$$u,v\in V^s, l\in L^{\gamma}, \gamma\in A_{rrive}(t),\ t\in\mathbb{T} \tag{23}$$

$$x_{u,v}^{l_b(f_i,f_{ib_k})}(t)\in\{0,1\},$$
$$u,v\in V^s, l_b\in L_b^{\gamma}, \gamma\in A_{rrive}(t),\ t\in\mathbb{T} \tag{24}$$

$$s_{c_j}^i = \left[S_{c_j}^i > 0\right], \quad c_j\in C, f_i\in\Phi \tag{25}$$

$$s_{c_j}^{ib_k} = \left[S_{c_j}^{ib_k} > 0\right], \quad c_j\in C, ib_k\in F_{ib}^{\gamma} \tag{26}$$

$$X^{\gamma}(t) = \min_{f_i\in\gamma}(\sum_{c_j\in C} x_{c_j}^{\gamma,i}(t))\in\{0,1\},$$
$$c_j\in C, i\in\Phi, \gamma\in A_{rrive}(t),\ t\in\mathbb{T} \tag{27}$$

Constraints (16) and (17) restrict a VNF (BVNF) only to be deployed on a single edge server c to avoid data dispersion. In constraint (18), $\lfloor\frac{r_f}{\delta}\rfloor$ represents maximum threshold of sharing number of f_i in any c during system execution time $\mathbb{T}$. When the threshold is reached, f_i running on c would experience fault due to overwork. At any time slot t, the constraint (19) indicates that computing resource required by all VNFs and BVNFs deployed on c cannot exceed remaining computing capacity of c. The constraint (20) indicates that reliability of receiving request γ must higher than reliability requirements of users. In addition, the $x_{c_j}^{\gamma,i}(t)$, $y_{c_j}^{\gamma,ib_k}(t)$, $z_{u,v}^{l(f_i,f_{i+1})}(t)$, $x_{u,v}^{l_b(f_i,f_{ib_k})}(t)$, $s_{c_j}^i$, $s_{c_j}^{ib_k}$, and $X^{\gamma}(t)$ are binary variables related to the service provisioning process.

IV. SFCs Approximation Deployment Algorithm

A. RES Overview

We propose a two-stage online scheme RES to solve the online LRSP problem, by jointly considering fluctuated Rliability, limit$E$d capacity, and the Sharing of VNFs and BVNFs. RES is consisted of two algorithm: SFCs Approximation Deployment (SAD) algorithm and REliability-Enhanced (REE) algorithm.

The SAD algorithm selects different VNFs to form SFCs requested by different user devices, and decides where to deploy the VNFs to minimize VNFs deployment and transmission costs. However, reliabilities of SFCs may not always higher than expectations of users because the VNFs are fault-prone under high workloads or are less reliable due to their own properties. Therefore, REE algorithm adaptively decides which VNFs need to be protected, and how much and where to deploy BVNFs to guarantee reliability while minimizing BVNFs deployment and synchronization costs. In addition, the scheme also reasonably reuses existing VNFs and BVNFs to further reduce deployment cost and improve resource utilization. Fig. 4 shows process of RES. In the second half of this section, SAD algorithm is expounded in detail, and then its feasibility and time complexity are analyzed.

B. SFCs Approximation Deployment Algorithm

Definition 3: Define H_i as critical transmission distance that communication cost of distance H_i is equal to deployment cost of instantiating f_i, and the c hosting f_i is taken as source point.

The SAD algorithm firstly searches for the nearest access point v from user devices requesting SFCs in $A_{rrive}(t)$ at each slot through the Dijkstra algorithm. However, access points with limited communication range are geographically distributed in the FP-MEC network. So, routing data from a user device to the nearest access point (called candidate

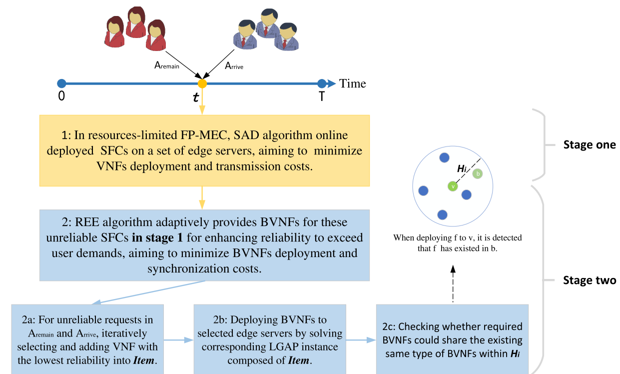


Fig. 4. An illustration of the RES.

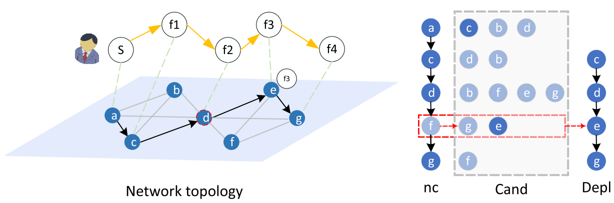


Fig. 5. An illustration of algorithm1.

Algorithm 1 SAD Algorithm

Input: $G^s = (V^s, E^s)$, $A_{rrive}(t)$, and S_c^i
Output: Update S_c^i, $A'_{rrive}(t)$

1 **begin**
2 $A'_{rrive}(t) = A_{rrive}(t)$;
3 **for** *each request* γ *in* $A'_{rrive}(t)$ **do**
4 Select nearest $v (v \in V^s)$ to γ;
5 **if** $v \notin C$ **then**
6 $v \leftarrow$ select nearest $c(c \in C)$ to v;
7 Initialize: $nc \leftarrow v$, $Depl = \varnothing$;
8 **for** *each VNF* f_i *in* γ **do**
9 $Cand = \varnothing$;
10 **if** $i > 1$ **then**
11 $nc \leftarrow c_b = \min\limits_{c_b \in C \backslash Depl} P_{Depl_{i-1}, c_b}$;
12 **if** $\exists\, a \in C \backslash Depl$ *within* H_i *and* nc *as source point* **then**
13 $Cand \leftarrow \forall a$;
14 **if** $\exists\, S_a^i > 0$ *in* $Cand \cup nc$ **then**
15 $Depl_i \leftarrow a' = \min\limits_{a' \in Cand} P_{nc, a'}$;
16 **else**
17 **if** $\exists\, cap_a(t) \geq cap_{f_i}$ *in* $Cand \cup nc$ **then**
18 $Depl_i \leftarrow a' = \min\limits_{a' \in Cand} P_{nc, a'}$;
19 **else**
20 $A'_{rrive}(t) \leftarrow A'_{rrive}(t) \backslash \gamma$, *break*;
21 Update S_c^i $(f_i \in \gamma,\ c \in C)$;
22 return $A'_{rrive}(t)$, S_c^i;

server nc), which must co-locate with edge server c, would go through multiple hops. Then, in the communication range H_i with nc as the source point, specific edge servers (e.g. a in line 12) are searched, and merged into the candidate set $Cand$. In addition, edge servers that have been selected to host $f_j \in \gamma$ $(0 < j < i)$ are recorded in the $Depl$ when f_i is deployed. Each VNF contained in a SFC should be distributed across different edge servers $c_b \in C \backslash Depl$, as shown in lines 11 and 12. This not only prevents all VNFs in a single server from fault due to the server failing, but also facilitates network workload balancing.

Finally, the f_i will be deployed on the selected a', which is the most suitable candidate server in nc and $Cand$. Specifically, the edge server that satisfies 1) already hosted f_i and 2) closest to nc has the highest priority in the selection process. For example, selecting node e (existing f_3 in e) to deploy f_3 instead of g and f node in Fig. 5. Although communication distances within H_i may increase, the overhead of creating a new VNF is avoided, thus reducing total receiving cost. Note that if no such server exists, we select the nearest server with sufficient remaining capacity. Otherwise, this implies that there are no suitable servers for hosting f_i at the slot. Therefore, capacities of occupied by VNFs $f_j \in \gamma$ $(0 < j < i)$ are revoked, and the SFC request is rejected.

VNFs contained in each γ are iteratively deployed to their corresponding suitable server a' based on service sequence demands. Intuitively, the number of existing VNFs climbs over time during service provisioning process, making it easier for new arrival requests to reuse and share common VNFs. Therefore, transmission cost among VNFs accounts for a larger proportion of total cost compared to VNFs deployment cost. In the algorithm 1, the edge servers with shorter communication distances from users are a major consideration, and deployment cost is used as an auxiliary judgment. Note that the sharing number of VNFs is limited. Exceeding threshold would incur faults. The value of threshold could be calculated based on history fault information and importance weight based on receiving cost of each VNF. The details of SAD is shown in algorithm 1.

C. Algorithm Analysis

In this section, we first prove the upper bound of total VNFs receiving cost C for A_{rrive} in a certain time slot. Then, the upper bound of difference between actual solution and theoretical optimal solution is proved. Finally, approximation ratio and time complexity of SAD are analyzed. To simplify the proof, we assume that F^γ is a constant, and all requests in A_{rrive} are pre-deploying at t slot, namely $A_{rrive} = A'_{rrive}$.

Lemma 1: In resource-limited FP-MEC networks, the upper bound of total VNFs receiving cost C for A_{rrive} obtained by algorithm 1 is:

$$C = \sum_{\gamma \in A_{rrive}} C_\gamma = (2 + \alpha_{max}) \sum_{\gamma \in A_{rrive}} \sum_{f_i \in \gamma} C^{depi}$$

where $\alpha_{max} = \dfrac{C_{tu} \cdot \max\limits_{P \in E^s} P}{C_{cu} \cdot \min\limits_{f \in \Phi} cap_f}$ *is a constant.* $P_{u,v}$ *is distances of the shortest physical path from* u *to* v $(u, v \in V^s)$. C^{depi} *is deployment cost of a VNF, and* C_i *is the sum of deployment and transmission cost of a VNF.*

Proof:

$$\begin{aligned} C_i &= C^{depi} + C^{tran}_{P_{Depl_{i-1}, nc_i}} + C^{tran}_{P_{nc_i, a'}} \\ &\leq 2C^{depi} + C^{tran}_{P_{Depl_{i-1}, nc_i}} \end{aligned} \tag{28}$$

$$\leq (2+\alpha)C^{depi}, \quad \alpha = \frac{C^{tran}_{P_{Depl_{i-1},nc_i}}}{C^{depi}} = \frac{C_{tu}P_{Depl_{i-1},nc_i}}{C_{cu}cap_{f_i}}$$

$$\leq (2+\alpha_{max})C^{depi}, \quad \alpha_{max} = \frac{C_{tu}\cdot \max\limits_{P\in E^s} P}{C_{cu}\cdot \min\limits_{f\in\Phi} cap_f} \tag{29}$$

The reason why formula (28) holds is that transmission cost between edge server nc and other candidate servers in $Cand$ is less than f_i deployment cost within communication range H_i.

$$C_\gamma = \sum_{i\in\gamma} C_i = (2+\alpha_{max})\sum_{i\in\gamma} C^{depi}$$

$\square$

Assuming deployment locations of requests in set $J \subseteq A_{rrive}$ is different from the optimal solution. Then, upper bound of difference between cost C_γ obtained by SAD and theoretical optimal cost C_{γ^*} is:

$$\sum_{\gamma\in A'_{rrive}}(C_\gamma - C_{\gamma^*}) = \sum_{\gamma\in J}(C_\gamma - C_{\gamma^*}) \leq \sum_{\gamma\in J} C_\gamma \leq \sum_{\gamma\in A'_{rrive}} C_\gamma \tag{30}$$

Theorem 3: Given a FP-MEC network with computing resource $cap_c(t)$, and new arrival requests $A_{rrive}(t) = \cup\gamma(S^\gamma, T^\gamma_a, T^\gamma_f, \mathrm{R}^\gamma, F^\gamma)$ at t slot, there is an efficient approximation SAD algorithm (algorithm 1) with an approximation ratio of $O((\frac{2}{\beta_{min}}+\theta')\theta+1)$ pre-deploying requested SFCs on the substrate network to solve sub-problem 1, aiming to minimize VNFs deployment and transmission cost. The time complexity of SAD is $O(|A_{rrive}(t)||F^\gamma||C|)$, where $\theta = \frac{\max_{f\in\Phi} cap_f}{\min_{f\in\Phi} cap_f}$, $\theta' = \frac{\max_{P\in E^s} P}{\min_{P\in E^s} P}$, $\beta_{min} = \frac{C_{tu}\cdot\min_{P\in E^s} P}{C_{cu}\cdot\min_{f\in\Phi} cap_f}$, cap_f is computing resource required by f, and P is link distances.

Proof: On the basis of Lemma 1 and formula (30), approximation ratio of algorithm 1 is proved as follows.

$$\frac{\sum\limits_{\gamma\in A'_{rrive}} C_\gamma}{\sum\limits_{\gamma\in A'_{rrive}} C_{\gamma^*}} = \frac{\sum\limits_{\gamma\in A'_{rrive}} C_\gamma - \sum\limits_{\gamma\in A'_{rrive}} C_{\gamma^*} + \sum\limits_{\gamma\in A'_{rrive}} C_{\gamma^*}}{\sum\limits_{\gamma\in A'_{rrive}} C_{\gamma^*}}$$

$$= \frac{\sum\limits_{\gamma\in A'_{rrive}} C_\gamma - C_{\gamma^*}}{\sum\limits_{\gamma\in A'_{rrive}} C_{\gamma^*}} + 1$$

$$\leq \frac{\sum\limits_{\gamma\in A'_{rrive}} C_\gamma - C_{\gamma^*}}{\sum\limits_{\gamma\in A'_{rrive}} |F^\gamma|\,(\min\limits_{i\in\Phi} C^{depi} + \min\limits_{P\in E^s} C^{tran}_P)} + 1$$

$$\leq \frac{\sum\limits_{\gamma\in A'_{rrive}} C_\gamma - C_{\gamma^*}}{\sum\limits_{\gamma\in A'_{rrive}} |F^\gamma|\,(1+\beta_{min})\min\limits_{i\in\Phi} C^{depi}} + 1,$$

$$\beta_{min} = \frac{C_{tu}\cdot\min\limits_{P\in E^s} P}{C_{cu}\cdot\min\limits_{f\in\Phi} cap_f}$$

$$\leq \frac{\sum\limits_{\gamma\in A'_{rrive}} (2+\alpha_{max})\,|F^\gamma|\max\limits_{i\in\Phi} C^{depi}}{\sum\limits_{\gamma\in A'_{rrive}} |F^\gamma|\,(1+\beta_{min})\min\limits_{i\in\Phi} C^{depi}} + 1$$

$$= \frac{(2+\alpha_{max})\max\limits_{i\in\Phi} C^{depi}}{(1+\beta_{min})\min\limits_{i\in\Phi} C^{depi}} + 1 \leq \frac{(2+\alpha_{max})\max\limits_{i\in\Phi} C^{depi}}{\beta_{min}\cdot\min\limits_{i\in\Phi} C^{depi}} + 1$$

$$= (\frac{2}{\beta_{min}}+\theta')\theta + 1, \quad \theta = \frac{\max\limits_{f\in\Phi} cap_f}{\min\limits_{f\in\Phi} cap_f}, \theta' = \frac{\max\limits_{P\in E^s} P}{\min\limits_{P\in E^s} P}$$

Time complexity analysis is as follows. Each VNF f_i needs to spend $O(|C|)$ time to select candidate servers, and check whether f_i has been deployed on them. The time consumed by other assignment operations is $O(1)$. For a request γ consisted of $|F^\gamma|$ VNFs, $|F^\gamma|$ selections are made. In addition, the maximum $O(|A_{rrive}(t)|)$ of deployment operations is performed for new arrival requests in t slot. Therefore, the time complexity of the algorithm 1 is $O(|A_{rrive}(t)|\,|F^\gamma|\,|C|)$. $\square$

V. Reliability-Enhanced Algorithm

A. Reliability-Enhanced Algorithm

After the first stage, there are a set of pre-deploying requests A'_{rrive} obtained by SAD algorithm, sharing VNFs number S^i_c, and a group of receiving requests A_{remain} that does not meet service time requirements before t slot. However, reliabilities of SFCs in A_{remain} and A'_{rrive} may not always higher than user expectations during service time. For this reason, additional BVNFs adaptively are deployed near VNFs in these SFCs.

Algorithm 2 REE Algorithm

Input: $G^s = (V^s, E^s)$, $A'_{rrive}(t)$, S^{ib}_c, $A_{remain}(t)$
Output: S^{ib}_c, $A_{ccept}(t)$, $A_{remain}(t)$

```
begin
    Item = ∅ ;
    if t > 1 & A_remain(t) ≠ ∅ then
        for each γ in A_remain(t) do
            while R̂^γ(t) < R^γ do
                i' ← min_{i∈γ} r̂^i(t), add f_{i'b} to selcted c_{j'}, update r̂^{i'}(t),
                R̂^γ(t);
    for each γ in A'_rrive(t) do
        while R^γ(t) < R^γ do
            Item ← f_{i'b} = min_{i∈γ} r^i(t), update r^{i'}(t), R^γ(t);
    Construct LGAP with Item and C in G^s;
    Solution S = ∪c_s is obtained by approximation algorithm [12];
    for each c_s ∈ S do
        if c_s > 0 then
            if ∃S^{ib_k}_{c_β} > 0 within H_i and c_s as source point then
                c_s ← c_{β'} = min P_{c_{β'},c_s};
        else
            A'_rrive(t) = A'_rrive(t)\γ;
    update S^{ib}_c, A_ccept(t) = A'_rrive(t), A_remain(t) = A_remain(t) ∪
    A_ccept(t);
    return S^{ib}_c, A_ccept(t), A_remain(t)
```

Firstly, ensure continuity of unreliable SFCs in received services in $A_{remain}(t)$. Iteratively add BVNFs near the least reliable VNFs in each SFC until their reliabilities again exceed user expectations, as shown in line 6. Specifically, we check whether there are BVNFs within $H_{i'}$, which are hosted on edge server $c_{j'}$ and have the same type as the lowest reliable $f_{i'}$. If they exist, select and share them to reduce BVNFs deployment cost. Otherwise, select an edge server $c_{j'}$ that has adequately computing capacity and is the closest to c_j hosting f_i to deploy BVNFs. Note that there is no additional BVNFs deployment cost to share existing BVNFs, but synchronization cost between VNFs and its BVNFs needs to be paid. In addition, lower reliable VNFs have a worse impact on SFC reliabilities, compared with higher reliable VNFs. Therefore, algorithm 2 prefers to provide backups for the unprotected VNFs, rather than protected VNFs.

Then, guarantee reliability of new arrival requests in $A'_{rrive}(t)$. Since most of new arrival SFCs are unreliable, we record the suitable numbers and type of backups in each SFC into $Item$ for optimal deployment. We reduce optimal BVNFs deployment problem to corresponding LGAP problem, aiming to maximize throughput of receiving requests while minimizing costs. The concrete construction process of LGAP instance is as follows. Each LGAP instance is consisted of a set of BVNFs (items) $\in Item$ with cost $C^{tsyn}_{l_b} + C^{depib}$ and size cap_{f_i}, and a set of edge servers (bins) with computing resource $cap_c(t)$. Through the approximation algorithm proposed by Cohen *et al.* [12], the solution $\mathbb{S} = \cup c_s$ of BVNFs deployment scheme could be preliminarily obtained, whose cost is no more than $2 - \epsilon$ $(0 < \epsilon < 1)$ times of optimal cost. In addition, we should check if there is the existing shareable f_{ib} within communication range H_i. Reusing f_{ib} can reduce the cost of each feasible solution c_s, and reserve more resources for the next slot. So, the receiving number of SFC requests during time horizon $\mathbb{T}$ would increase.

B. RES Scheme

The detailed process of approximation RES scheme with an approximation ratio of $O((\dfrac{2}{\beta_{min}} + \theta')\theta + 1)$ is as follows. At begin of each slot, reliability of f_i $(i \in \Phi)$ on different edge servers is first updated based on formula (3) and S^i_c. Then, the LRSP problem in resource-limited FP-MEC is solved by executing algorithms 1 and 2 in real-time. Finally, resources occupied by users who dynamical departure at this slot are recovered, and S^{ib}_c, S^i_c are updated. The execution process is limited by time span $\mathbb{T}$.

Algorithm 3 RES Scheme

Input: $G^s = (V^s, E^s)$, $\mathbb{T}$, and $A_{rrive}(t)$
Output: Reliability-enhanced SFCs provisioning scheme
begin
  for *each* $t \in \mathbb{T}$ **do**
    Updated r^i_c according to formula (3) and S^i_c;
    $A'_{rrive}(t)$ is obtained by Algorithm 1 with input G^s, $A_{rrive}(t)$, and S^i_c;
    $A_{accept}(t)$ and $A_{remain}(t)$ are obtained by Algorithm 2 with input G^s, $A'_{rrive}(t)$, and S^{ib}_c;
    for *each* γ *in* $A_{remain}(t)$ **do**
      if $T^\gamma_f = t$ **then**
        $A_{remain}(t) = A_{remain}(t) \backslash \gamma$;
    Update S^{ib}_c, S^i_c, $A_{remain}(t+1) = A_{remain}(t)$;

C. Algorithm Analysis

In this section, we analyze feasibility, time complexity and approximation ratio of REE algorithm (algorithm 2) and RES scheme (algorithm 3).

Theorem 4: Given a FP-MEC network composed of edge servers C with computing resource cap_c, access points V, physical link E^s, a set of requests $A'_{rrive}(t)$ obtained by algorithm 1, and a group of receiving requests $A_{remain}(t)$ that does not meet service time requirements before t slot, there is an approximation algorithm 2 with $2 - \epsilon$ $(0 < \epsilon < 1)$ approximation ratio provides a feasible solution in $O((|C|\, log\,|A'_{rrive}| + 2\,|C| + |F^\gamma|)\,|A'_{rrive}| + |F^\gamma|\,|A_{remain}|\,|C|)$ time for the sub-problem 2, where $|C|$ is the total number of edge servers and $|F^\gamma|$ is the maximum number of VNFs to form a single SFC in all requests.

Proof: First, prove algorithm 2 can provide a feasible solution. For unreliable requests in A_{remain}, redundant BVNFs f_{ib} could be shared or instantiated on the nearest edge server c with sufficient remaining computing capacity. For unreliable requests in A'_{rrive}, capacities of server are guaranteed not to violate during processes of solving LGAP instance. In other words, f_{ib} is deployed only when resource required by f_{ib} is less than or equal to remaining resource of c. Therefore, BVNFs deployment solution based on algorithm 2 is feasible. Then, approximation ratio of REE is proved as follows.

$$\begin{aligned}
C_{IncSall} &= \sum_{\gamma \in A'_{rrive}} C_{IncS} + \sum_{\gamma \in A_{remain}} C_{IncS} \\
&= \sum_{\gamma \in A'_{rrive}} C^{dep}_{\gamma b} + C^{tsyn}_{\gamma} + \sum_{\gamma \in A_{remain}} C_{IncS^*} \\
&\le (2-\epsilon) \sum_{\gamma \in A'_{rrive}} C^{dep}_{\gamma b^*} + C^{tsyn}_{\gamma^*} + \sum_{\gamma \in A_{remain}} C_{IncS^*} \\
&\le (2-\epsilon)\Big(\sum_{\gamma \in A'_{rrive}} C_{IncS^*} + \sum_{\gamma \in A_{remain}} C_{IncS^*} \Big) \\
&= (2-\epsilon) C_{IncSall^*}
\end{aligned}$$

BVNFs receiving cost C_{IncS^*} of A_{remain} is always optimal by enumeration, and BVNFs receiving cost of A'_{rrive} is no more than $2 - \epsilon$ $(0 < \epsilon < 1)$ times of the optimal cost [12]. Finally, time complexity is analyzed. To ensure continuity and reliability of services in A_{remain}, checking reliability of each γ and iteratively providing BVNFs for the lowest reliable f_i within H_i takes $O(|A_{remain}|\,|F^\gamma|\,|C|)$. It takes time $O(|A'_{rrive}|\,|F^\gamma|)$ to merge f_i with lower reliability in A'_{rrive} into $Item$, and construct $Item$ as a LGAP instance takes time $O(|A'_{rrive}|\,|C|)$. The approximation solution $\mathbb{S}$ is obtained, and optimize the solution $\mathbb{S}$ take $O(|A'_{rrive}|\,|C|\, log\,|A'_{rrive}|)$ [12] and $O(|A'_{rrive}|\,|C|)$ time respectively. Therefore, the total time complexity of algorithm 2 is $O((|C|\, log\,|A'_{rrive}| + 2\,|C| + |F^\gamma|)\,|A'_{rrive}| + |F^\gamma|\,|A_{remain}|\,|C|)$. □

Theorem 5: Given time span $\mathbb{T}$, FP-MEC network $G^s = (V^s, E^s)$ with computing resource cap_c, and a set of requests A_{rrive} that arbitrary departure or arrival at begin of each slot. There is an approximation scheme RES with an approximation ratio of $O((\frac{2}{\beta_{min}} + \theta')\theta + 1)$ for LRSP problem, whose time complexity is $O((log\,|A_{rrive}| + |F^\gamma|)\,|C|\,|A_{rrive}|)$, where, $\theta = \frac{\max_{f\in\Phi} cap_f}{\min_{f\in\Phi} cap_f}$, $\theta' = \frac{\max_{P\in E^s} P}{\min_{P\in E^s} P}$, $\beta_{min} = \frac{C_{tu}\cdot \min_{P\in E^s} P}{C_{cu}\cdot \min_{f\in\Phi} cap_f}$, $|C|$ is the total number of servers and $|F^\gamma|$ is the maximum number of VNFs to form a single SFC in all requests.

Proof: The approximation ratio of algorithm 3 is proved as follows.

$$
\begin{aligned}
C_{all} &= C_{DepS} + C_{IncS} \\
&\le ((\frac{2}{\beta_{min}} + \theta')\theta + 1)C_{DepS^*} + (2-\epsilon)C_{IncS^*} \\
&\le ((\frac{2}{\beta_{min}} + \theta')\theta + 1)(C_{DepS^*} + C_{IncS^*}) \\
&= ((\frac{2}{\beta_{min}} + \theta')\theta + 1)C_{all^*} \qquad (31)
\end{aligned}
$$

The reason why formula (31) holds is:

$$
\begin{aligned}
\theta &= \frac{\max_{f\in\Phi} cap_f}{\min_{f\in\Phi} cap_f}(\theta' = \frac{\max_{P\in E^s} P}{\min_{P\in E^s} P}) \ge 1 \\
&\Rightarrow (\frac{2}{\beta_{min}} + \theta')\theta + 1 \ge 2 \\
&\Rightarrow (\frac{2}{\beta_{min}} + \theta')\theta + 1 \ge 2-\epsilon \ (0<\epsilon<1)
\end{aligned}
$$

The time complexity of RES scheme is analyzed as follows. At begin of each slot, it takes $O(|\Phi|\,|C|)$ time to update the reliability of each type VNFs on different edge servers based on S_c^i and formula (3). Algorithm 1 and algorithm 2 take $O(|A_{rrive}|\,|F^\gamma|\,|C|)$ and $O((|C|\,log\,|A'_{rrive}| + 2\,|C| + |F^\gamma|)\,|A'_{rrive}| + |F^\gamma|\,|A_{remain}|\,|C|)$, respectively. At the end of each slot, it takes time $O(|A_{ccept}|)$ to detect if there are any requests to leave and reclaim their resources. As a result, the total time complexity of algorithm 3 is $O((log\,|A_{rrive}| + |F^\gamma|)\,|C|\,|A_{rrive}|)$. □

VI. Performance Evaluation

In this section, we first describe experimental environments and benchmark schemes for LRSP problem. Then, different scheme performances are evaluated and important parameter influences are analyzed based on the real network topology of CERNET in China. All experiments are run on a personal computer with 2.30GHz Intel(R) Core(TM) I7-10875 CPU and 16GB of RAM.

A. Environment Setting

We conduct simulation experiments in MATLAB based on the real network topology of the China Education and Research Network (CERNET) consisting of 37 physical nodes.

Fig. 6. The CERNET.

The locations of these nodes are consistent with the actual latitude and longitude. The CERNET is used as the FP-MEC substrate network $G^s = (V^s, E^s)$ and shown in Fig. 6. In order to deploy Firewall, NAT, IDS and other different types of VNFs [35] into edge servers, we randomly configure edge servers for 50% network size physical nodes. The computing capacities of each edge server range from 2000 MHz to 4000 MHz [3]. The number of VNF type Φ is set to 15, and each VNF has computing demand range form 10 MHz to 100 MHz and its initial reliability range of 0.85 to 0.99 [4].

In addition, each SFC request has a specific sequence, consisting of four random VNFs. These requests location-randomly appear in G^s and serve users for 1-3 consecutive time slots. The reliability demands of SFCs ranges from 0.7 to 0.99. The reliability of BVNFs that require same computing capacity as VNFs is set to 0.9. The proportionality coefficient δ between increment of fault probability ρ_{f_i} and sharing number of f_i is set to 0.02. The bandwidth level of FP-MEC network is 1 ($B_e = 1$ MHz) [36]. $C_{cu} = 0.03375$, $C_{tu} = 0.085$ [10]. The above values are default values for each trial unless otherwise specified. In addition, all data in figures are averages of 100 trials.

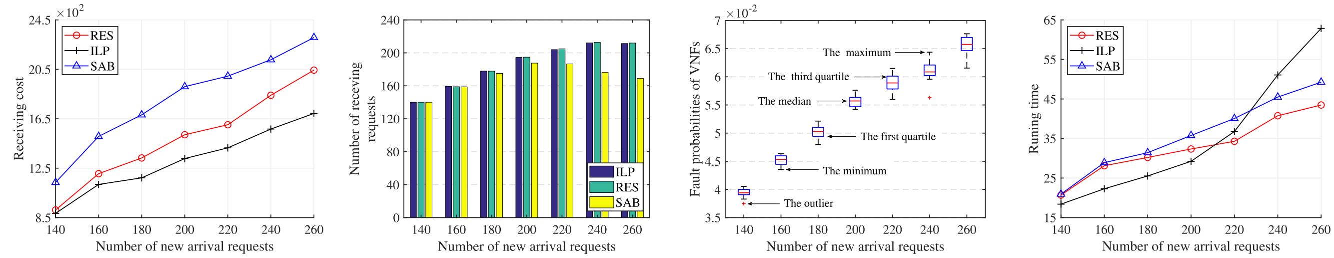
B. Benchmark

In this section, we compare performances of RES scheme with following schemes. Note that ILP and SAB schemes are also implemented by sharing VNFs and BVNFs, both of which have lower receiving costs than original schemes.

RES scheme: In order to solve the LRSP problem, this paper proposes RES scheme.

SAB scheme: In [18], authors proposed a SAB scheme that combines static and dynamic BVNFs deployments. They first provided a BVNF for each VNFs in arrival requests. Then, BVNFs were provided dynamically in real time for unreliable requests.

ILP scheme: Solving the LRSP problem directly with ILP solver is a time-consuming and laborious task. Therefore, we first preliminarily relax deployment-related decision variables to non-integer. Then, On the basis of relaxation solutions, the decision variable of edge server that is most suitable (e.g.,



(a) Receiving cost (b) Receiving requests number (c) Fault probabilities of VNFs (d) Running time

Fig. 7. Performances of different schemes by varying arrival requests number from 140 to 260.

fractional value of this variable is maximum among all servers) for deploying f is set to 1. Similar operations are performed on other decision variables. Finally, check computing capacities of each edge server. If capacity violations occur, iteratively remove SFCs deployed on the server with minimal ratio of reliability to computing resource until no violations. If capacity surpluses occur, try to iteratively deploy a unreceived request with maximum ratio of reliability to resource to servers. However, there may be no solution when providing VNFs and BVNFs simultaneously for large-scale requests due to limited capacities of edge servers after the preliminarily linear relaxation program (LRP). Therefore, we further relaxed arrival number of requests to avoid this, where arrival request number participating in LRP is less than actual arrival number of requests. This means that a request with the lowest ratio of reliability to computing capacity would be rejected when there is no solution, and then continues to execute LRP. Above relaxed operations are repeated until a solution is found. The ILP scheme not only provides a lower bound on receiving cost for all schemes, but also receives as many new arrival requests as possible. Therefore, we regard the ILP scheme as the optimal solution compared with RES and SAB.

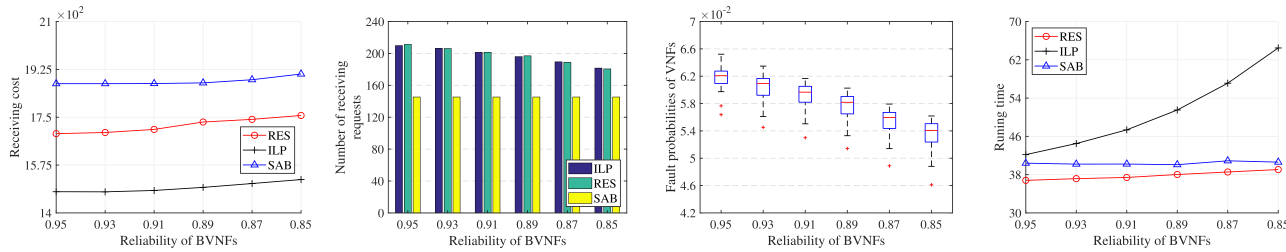
C. Performance Evaluation

1) Impact of the Number of New Arrival Requests on Performances of Different Schemes: We vary the number of arrival requests per slot from 140 to 260, and set reliability demands ranging from 0.75 to 0.85. Fig. 7(a) shows that receiving cost increases with request numbers, where cost of RES is no more than 1.20 times of cost-optimal ILP, and the cost is only 88.56% of SAB when request numbers are 260. Fig. 7(b) exhibits throughput of two trends that three schemes increase gradually first and then remain stable or decline when arrival number is around 220. The reason is that computing capacity of each edge server is limited, so receiving request number is also limited. In addition, the fluctuation of throughput after leveling off due to dynamic departure or arrival of users. Fig. 7(b)(c) demonstrate that the throughput of RES is positively correlated with VNFs fault probability in each slot. This is same as Theorem 1. Notice that time complexity of ILP is exponential time, which means that ILP scheme does not apply to large scale and real-time provisioning problems. Form 7(d), we can be observed RES as a promising scheme to provide approximation solutions with polynomial time complexity.

2) The Impact of BVNFs Reliability on Performances of Different Schemes: We reduce the expectation of all BVNF reliabilities from 0.95 to 0.85, and set reliability demands ranging from 0.7 to 0.8. Fig. 8(a) exhibits that receiving cost slowly climbs with BVNF reliability decreases, where cost of RES is no more than 1.18 times of ILP and is only 90% of SAB. This is because that more BVNFs need to be deployed to exceed user reliability expectations as their reliability deteriorates. The increasing number of BVNFs not only adds deployment and synchronization cost, but also rises resource competition for the next slot due to resource-constrained capacities, which means that receiving number would decrease, as shown in Fig. 8(b). The SAB remains stable because the competition of remaining resource is always maximized due to it provides lots of backups to all VNFs. Fig. 8(c) demonstrates that VNF fault probabilities drop as throughput decreases due to reduced BVNFs reliability. Fig. 8(d) displays that running time of ILP increases exponentially, and RES lower than SAB and ILP.

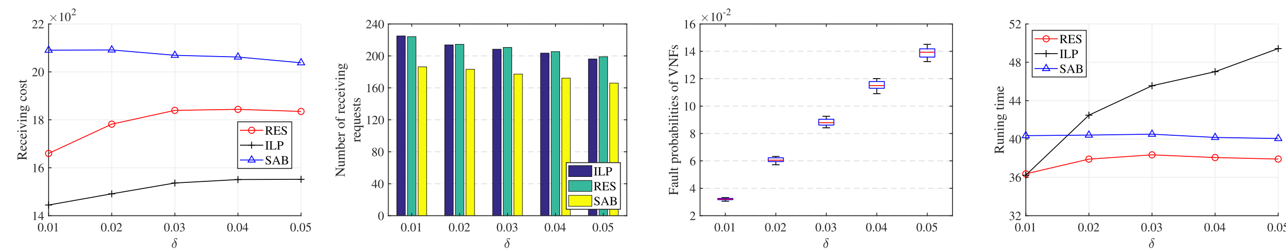
3) The Impact of δ on Performances of Different Schemes: We vary the proportional coefficient δ that is a fraction of increment of ρ_{f_i} to number sharing of f_i from 0.01 to 0.05, and set reliability demands ranging from 0.7 to 0.8. Fig. 9(a) shows that receiving cost rises first and then reduces with increase of δ. The reason is that VNFs have to be deployed to servers far from users due to growing workload sensitivity δ, increasing communication cost. The reason for the reduction is confirmed by Fig. 9(b), where receiving requests drop as δ increases, reducing total receiving cost of requests. Fig. 9(b)(c) demonstrate an interesting phenomenon that VNFs fault probability climbs when total throughput declines, which because the probability is not only related to sharing request number, but also affected by δ. During the experiment, growth of δ has a greater influence on fault probability than reduction of receiving request numbers. As can be seen from Fig. 9(d), RES always has the lowest running time, and gap between it and SAB remains stable, while ILP running time increases with δ.

4) The Impact of VNF Number in SFC on Performances of Different Schemes: We add VNF number in a single SFC from 4 to 20, and set the arrival request number to 100. From Fig. 10(a), costs of ILP and RES rise with the VNF number, while SAB is abnormal at 20, even with 0 cost. This is due to limitations of SAB, which provides a large amount of backup for every VNF whether reliable or not, resulting in far fewer requests being accepted than RES and ILP in Fig. 10(b), 7(b).



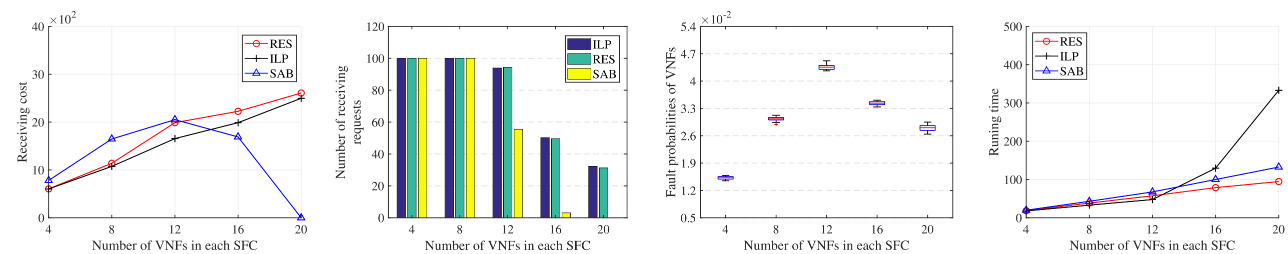
(a) Receiving cost (b) Receiving requests number (c) Fault probabilities of VNFs (d) Running time

Fig. 8. Performances of different schemes by reducing BVNFs reliability from 0.95 to 0.85.



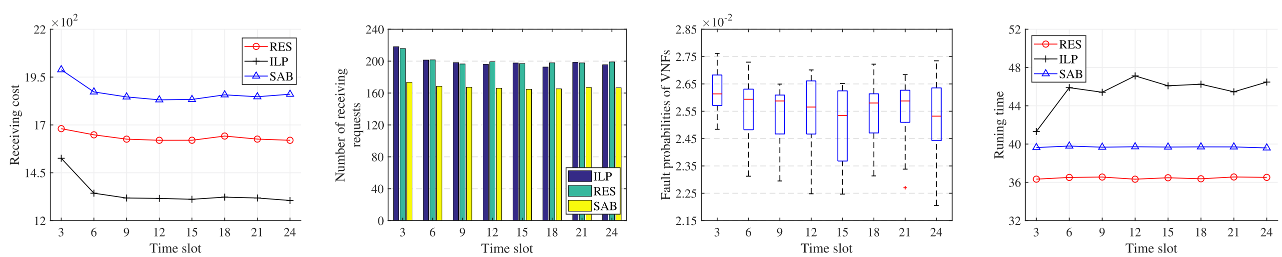
(a) Receiving cost (b) Receiving requests number (c) Fault probabilities of VNFs (d) Running time

Fig. 9. Performances of different schemes by increasing the δ from 0.01 to 0.05.



(a) Receiving cost (b) Receiving requests number (c) Fault probabilities of VNFs (d) Running time

Fig. 10. Performances of different schemes by increasing VNF number in SFC from 4 to 20.



(a) Receiving cost (b) Receiving requests number (c) Fault probabilities of VNFs (d) Running time

Fig. 11. Performances of different schemes by increasing the slot from 3 to 24.

Fig. 10(d) demonstrates that running time of RES is only 20% of ILP when VNFs and BVNFs are deployed on a large scale, that is, each SFC contains 20 VNFs and more BVNFs. Note that ILP and SAB are difficult to solve dynamic and real-time servers provisioning problem, because running time of ILP and consuming capacities of SAB are intolerable.

5) The Impact of Other Parameters on Performances of Different Schemes: We study effect of time span $\mathbb{T}$ and number of VNF types Φ on performances, respectively, by changing $\mathbb{T}$ from 3 to 25, and varying Φ from 15 to 35. As can be seen from Fig. 11, 12, these parameters have little impact on performances of different schemes except receiving cost in Fig. 11(a) and VNF fault probabilities in Fig. 12(c). In terms of the cost, the initial phase is expensive because no existing VNFs and BVNFs can be shared. In terms of the fault probabilities, the frequency of use of any VNF would drop as VNF types rise, so workloads of each VNF also decrease.

6) The Impact of Some Assumptions on Performances of RES: We first explore the impact of bandwidth capacities by varying available bandwidth levels from 1 to 5, setting new arrival request number is 150, randomly extracting bandwidth demands of users within $[1, 4]$. From Fig. 13, the change of performance gaps between RES and optimal ILP keeps stable, which means bandwidth capacities have little influence on accuracy of RES scheme. Then, we explore the impact between using BVNF reliability expectation and using different BVNF reliabilities that same as its VNF reliabilities. From Fig. 14, using the expectation could reduce running

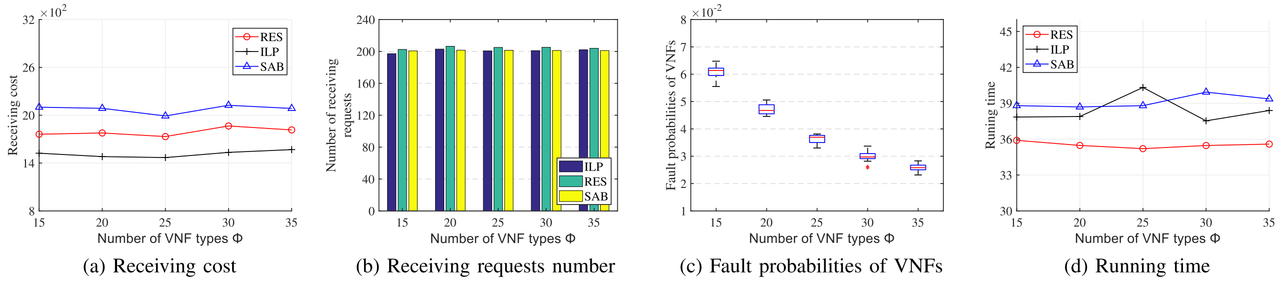


Fig. 12. Performances of different schemes by increasing VNF type number Φ from 15 to 35.

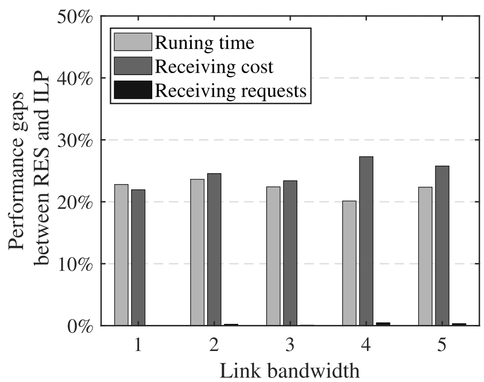


Fig. 13. Performance gaps on link bandwidth.

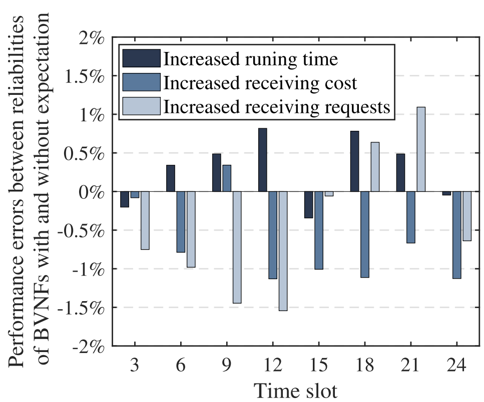


Fig. 14. Performance errors on BVNF reliabilities.

time of RES, but incur a maximum relative error of 1.51% on receiving cost and number.

VII. Conclusion

In this paper, we first study the LRSP problem from a new perspective that considers fault-prone nature of VNFs and the sharing of VNFs and BVNFs, so as to maximize throughput of receiving requests while minimizing receiving cost. Then, the problem is discretized into a sequence of ORSP problems, where both kinds of problem are proved to be NP-hard. To reduce the hardness, a two-stage online scheme RES with an approximation ratio of $O((\frac{2}{\beta_{min}}+\theta')\theta+1)$ is proposed to solve ORSP by trading-off among limited capacity, fluctuated reliability, and the sharing of VNFs and BVNFs. In the first stage, SAD algorithm selects different VNFs to form requested SFCs. In the second stage, REE algorithm adaptively provides BVNFs near VNFs by solving a series of LGAP instance for SFCs whose reliabilities are lower than user expectations. Finally, extensive theoretical analyses and experiments based on real network topology of CERNET in China demonstrate that RES scheme could reduce receiving cost by at least 10% and effectively adapt to fault-prone networks.

References

[1] S. Xia, Z. Yao, Y. Li, and S. Mao, "Online distributed offloading and computing resource management with energy harvesting for heterogeneous MEC-enabled IoT," *IEEE Trans. Wireless Commun.*, vol. 20, no. 10, pp. 6743–6757, Oct. 2021.

[2] 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.

[3] 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.

[4] 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.

[5] P. Gill, N. Jain, and N. Nagappan, "Understanding network failures in data centers: Measurement, analysis, and implications," in *Proc. ACM SIGCOMM Conf.*, 2011, pp. 350–361.

[6] R. Govindan, I. Minei, M. Kallahalla, B. Koley, and A. Vahdat, "Evolve or die: High-availability design principles drawn from Googles network infrastructure," in *Proc. ACM SIGCOMM Conf.*, Aug. 2016, pp. 58–72.

[7] D. Li, P. Hong, K. Xue, and J. Pei, "Availability aware VNF deployment in datacenter through shared redundancy and multi-tenancy," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 4, pp. 1651–1664, Dec. 2019.

[8] B. Han, V. Gopalakrishnan, G. Kathirvel, and A. Shaikh, "On the resiliency of virtual network functions," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 152–157, Jul. 2017.

[9] Q. Lin *et al.*, "Predicting node failure in cloud service systems," in *Proc. 26th ACM Joint Meeting Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.*, Oct. 2018, pp. 480–490.

[10] Y. Ma, W. Liang, M. Huang, W. Xu, and S. Guo, "Virtual network function service provisioning in MEC via trading off the usages between computing and communication resources," *IEEE Trans. Cloud Comput.*, early access, Dec. 8, 2021, doi: 10.1109/TCC.2020.3043313.

[11] R. M. Nauss, "Solving the generalized assignment problem: An optimizing and heuristic approach," *INFORMS J. Comput.*, vol. 15, no. 3, pp. 249–266, Aug. 2003.

[12] R. Cohen, L. Katzir, and D. Raz, "An efficient approximation for the generalized assignment problem," *Inf. Process. Lett.*, vol. 100, no. 4, pp. 162–166, 2006.

[13] S. Herker *et al.*, "Survey on survivable virtual network embedding problem and solutions," in *Proc. Int. Conf. Netw. Services (ICNS)*, 2013, pp. 99–104.

[14] D. Cotroneo *et al.*, "Network function virtualization: Challenges and directions for reliability assurance," in *Proc. IEEE Int. Symp. Softw. Rel. Eng. Workshops*, Nov. 2014, pp. 37–42.

[15] L. Qu, C. Assi, K. Shaban, and M. J. Khabbaz, "A reliability-aware network service chain provisioning with delay guarantees in NFV-enabled enterprise datacenter networks," *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 3, pp. 554–568, Sep. 2017.

[16] J. Fan, M. Jiang, O. Rottenstreich, Y. Zhao, T. Guan, R. Ramesh, S. Das, and C. Qiao, "A framework for provisioning availability of NFV in data center networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 2246–2259, Oct. 2018.

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

[18] 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,” in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Jul. 2020, pp. 2096–2105.

[19] S. Yang, F. Li, R. Yahyapour, and X. Fu, “Delay-sensitive and availability-aware virtual network function scheduling for NFV,” *IEEE Trans. Services Comput.*, vol. 15, no. 1, pp. 188–201, Jan. 2022.

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

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

[22] 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.

[23] 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.

[24] 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. (INFOCOM)*, Apr. 2019, pp. 1918–1926.

[25] W. Ding, H. Yu, and S. Luo, “Enhancing the reliability of services in NFV with the cost-efficient redundancy scheme,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[26] N. T. Dinh and Y. Kim, “An efficient reliability guaranteed deployment scheme for service function chains,” *IEEE Access*, vol. 7, pp. 46491–46505, 2019.

[27] J. Fan, M. Jiang, and C. Qiao, “Carrier-grade availability-aware mapping of service function chains with on-site backups,” in *Proc. IEEE/ACM 25th Int. Symp. Quality Service (IWQoS)*, Jun. 2017, pp. 1–10.

[28] M. Huang, W. Liang, Y. Ma, and S. Guo, “Maximizing throughput of delay-sensitive NFV-enabled request admissions via virtualized network function placement,” *IEEE Trans. Cloud Comput.*, vol. 9, no. 4, pp. 1535–1548, Oct. 2021.

[29] 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.

[30] J. L. Rodgers and W. A. Nicewander, “Thirteen ways to look at the correlation coefficient,” *Amer. Statist.*, vol. 42, no. 1, pp. 59–66, 1988.

[31] B. Schroeder and G. A. Gibson, “A large-scale study of failures in high-performance computing systems,” *IEEE Trans. Depend. Sec. Comput.*, vol. 7, no. 4, pp. 337–350, Oct./Dec. 2009.

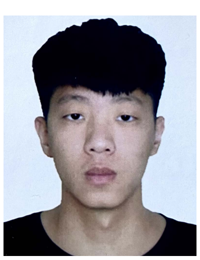
[32] Y. Sharma, W. Si, D. Sun, and B. Javadi, “Failure-aware energy-efficient VM consolidation in cloud computing systems,” *Future Gener. Comput. Syst.*, vol. 94, pp. 620–633, May 2019.

[33] Y. Ma, W. Liang, and S. Guo, “Mobility-aware delay-sensitive service provisioning for mobile edge computing,” in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2019, pp. 270–276.

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

[35] A. De Domenico, Y.-F. Liu, and W. Yu, “Optimal virtual network function deployment for 5G network slicing in a hybrid cloud infrastructure,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 12, pp. 7942–7956, Dec. 2020.

[36] Z. Yang, C. Pan, K. Wang, and M. Shikh-Bahaei, “Energy efficient resource allocation in UAV-enabled mobile edge computing networks,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4576–4589, Sep. 2019.



Yu Qiu received the B.E. degree from the Tianjin University of Technology, Tianjin, China, in 2020. He is currently pursuing the M.S. degree in electronic and information engineering with Guangxi University, Nanning, China. His research interests include mobile edge cloud, network function virtualization, and optimization theory.

Junbin Liang received the B.E. and M.S. degrees from Guangxi University, Nanning, China, in 2000 and 2005, respectively, and the Ph.D. degree from Central South University, Changsha, China, in 2010. He was a Visiting Professor with The University of British Columbia from 2019 to 2020. He is currently a Professor at Guangxi University. His research interests include sensor-cloud systems, fog computing, and distributed computing.

Victor C. M. Leung (Life Fellow, IEEE) is a fellow of the Royal Society of Canada (Academy of Science), Canadian Academy of Engineering, and Engineering Institute of Canada. He has received the 1977 APEBC Gold Medal, the 1977–1981 NSERC Postgraduate Scholarships, the IEEE Vancouver Section Centennial Award, the 2011 UBC Killam Research Prize, the 2017 Canadian Award for Telecommunications Research, the 2018 IEEE TCGCC Distinguished Technical Achievement Recognition Award, and the 2018 ACM MSWiM Reginald Fessenden Award. He has coauthored papers that won the 2017 IEEE ComSoc Fred W. Ellersick Prize, the 2017 IEEE SYSTEMS JOURNAL Best Paper Award, the 2018 IEEE CSIM Best Journal Paper Award, and the 2019 IEEE TCGCC Best Journal Paper Award. He is serving on the editorial boards for the IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, IEEE TRANSACTIONS ON CLOUD COMPUTING, IEEE ACCESS, *IEEE Network*, and several other journals. He is named in the current Clarivate Analytics list of “Highly Cited Researchers.”

Xu Wu worked as a Visiting Scholar with the School of Engineering and Technology, Indiana University–Purdue University Indianapolis, Indianapolis, USA. She is currently an Associate Professor at the School of Computer, Electronics and Information, Guangxi University. Her research interests include trusted computing, pervasive computing, mobile computing, and software engineering.

Xia Deng (Member, IEEE) joined the School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou, China, where she is currently an Assistant Professor. Her current research interests include LEO satellite networks, mobile social networks, and protocol design and optimization in mobile networks. She has received the Best Paper Award at IEEE ICC 2013.