Journal of Network and Systems Management (2023) 31:18
https://doi.org/10.1007/s10922-022-09708-x

Efficient Redundancy Allocation for Reliable Service Function Chains in Edge Computing

Thi-Thuy-Lien Nguyen[1,2,3] · **Tuan-Minh Pham**[1,2] · **Linh Manh Pham**[3]

Received: 25 May 2022 / Revised: 27 September 2022 / Accepted: 6 December 2022

Abstract

Ensuring the high reliability of service function chains (SFCs) in Edge Computing, in which several distributed edge servers are available, is a challenging issue. Previous studies on reliable SFCs ignore the impact of physical hardware failures when multiple virtual network functions (VNFs) are deployed on the same server, resulting in inaccurate reliability estimates. In this paper, we first propose an optimization model and approximation algorithm, considering both hardware and software reliability, to maximize the reliability of SFCs in each service demand. We then develop an algorithm to increase the reliability of SFCs to a given requirement. The evaluation results show that our algorithms achieve a near-optimal solution with a significant reduction in the computational time for finding the placement of redundant VNFs. We also observe that our proposed redundancy VNF allocation can efficiently save the backup cost to achieve a given SFC reliability requirement.

Keywords Edge computing · NFV · VNF · Redundancy · Reliability

✉ Thi-Thuy-Lien Nguyen
lien.nguyenthithuy@phenikaa-uni.edu.vn

✉ Tuan-Minh Pham
minh.phamtuan@phenikaa-uni.edu.vn
https://sites.google.com/site/phamtuanminh/

Linh Manh Pham
linhmp@vnu.edu.vn

1 Faculty of Computer Science, Phenikaa University, Hanoi 12116, Vietnam

2 Phenikaa Research and Technology Institute (PRATI), A&A Green Phoenix Group JSC, No. 167 Hoang Ngan, Trung Hoa, Cau Giay, Hanoi 11313, Vietnam

3 Faculty of Information Technology, VNU University of Engineering and Technology, Hanoi 100000, Vietnam

Published online: 19 December 2022

1 Introduction

Ensuring the high reliability of service function chains (SFCs) in Edge Computing, in which several distributed edge servers are available, is one of the big and essential challenges for assuring future survival and growth of the network functions virtualization (NFV) systems. In NFV-enabled systems, a network service is composed of some smaller virtual network functions (VNFs) deployed dispersedly in the networks. Dispersion of services gives many benefits such as cost reduction, flexible resource management [1], however, failures in one VNF may lead to a collapse of the entire SFC. Therefore, the NFV-based networks indicated higher reliability requirements than the conventional networks, especially in Edge Computing where storage and processing bandwidth of the edge layer are limited. As a result, a high reliability guarantee for service requests over Edge Computing is an important challenge to be solved.

The reliability concepts and requirements are provided by European Telecommunications Standards Institute (ETSI) in [2]. The reliability of a service in NFV-enabled systems depends on the VNFs of the service and the servers that the VNFs are placed on. According to previous studies on reliable SFCs, there are two main ways to improve the reliability of SFCs in NFV-enabled systems. A simple approach is VNFs placement optimizations. Most of the proposed methods focus purely on original VNFs placement schemes by selecting a substrate node with high reliability to deploy VNFs [3–5]. Its drawbacks are not enough to ensure adequately high reliability for services or to avoid hardware failures. Another common approach to improve the reliability of SFCs is VNFs redundant deployments [6–9]. It is referred to the allocation of additional VNF copies to SFCs. Deploying VNFs redundant is effective in ensuring high reliability of services but it requires extra resources resulting in increased costs. A VNF may need multiple backup copies to improve its reliability, so simply duplicating each VNF with a fixed number of backup instances may exceed the edge resource capacity or require unnecessary hardware expense. Therefore, it is necessary to minimize the cost of redundant deployment while adequately ensuring the reliability requirements of the services.

The redundancy allocation cost minimization problem becomes more and more complicated, especially in the Edge Computing, because of the resource limitations of the edge. Some previous researches consider the reliability problem of end-to-end network chains in the edge networks in [10–12]. These studies protect unreliability VNFs on the basis of considering software or hardware failures separately and place these redundancies on the system. They ignore the impact of the reliability of physical hardware on SFCs' reliability when multiple VNFs are deployed on the same server, resulting in inaccurate reliability estimates. Other pieces of research consider both functional reliability and underlying hardware reliability for the reliability guarantee problem in [13–15]. However, the method proposed in [13] may not take over in case of single edge server failures, and [14, 15] assume that each service demand only requests one VNF instance while we consider NFV services consisting of several VNFs.

To the best of our knowledge, little attention has been given to optimize resource consumption for the VNF's redundant deployment on the basis of considering both hardware and software reliability in the Edge Computing. The main contributions of the paper are as follows:

- We propose a mixed integer linear programming (MILP) model and an approximate algorithm that allow finding the optimal solution and an approximate solution for the joint optimization placement problem of primary VNFs and their corresponding backup VNFs in the edge layer of an NFV-enabled IoT (NIoT) system. On the one hand, the proposed model ensures the tolerance of SFCs that can take over if any single node is corrupted in the edge layer. On the other hand, it finds a sweet spot between the deployment costs and the minimal reliability of SFCs overall service requests on the basis of considering simultaneously hardware and software reliability.
- We then formulate a VNFs redundancy allocation cost minimization problem and propose a cost-efficient VNFs redundancy scheme. The proposed solutions allow improving the reliability of SFCs to meet predefined requirements of IoT services over the Edge Computing on the basis of considering simultaneously software and hardware reliability and VNFs' resource consumption.
- In our experiments, the results show that our algorithms achieve the near-optimal solution with a significant reduction in the computational time in finding a placement solution of original VNFs and their full-backup VNFs. Our experiments also show that the proposed VNFs selection scheme for backup deployment performs well for saving the backup cost for up to 30–40% with the same ratio of the satisfied service requests.

The remainder of this paper are structured as follows. In Sect. 2, we show our short review of related work. In Sect. 3, we present a brief background for the reliability models of NFV network services and our research motivation. Next, we describe an NFV-enabled IoT system in the Edge Computing in Sect. 4. We first propose a MILP model to achieve the optimal solution for the joint optimization placement problem of primary VNFs and their corresponding backup VNFs in the edge layer of the NIoT system in Sect. 5. We then propose an approximate algorithm that allows finding an approximate solution in a large-scale NIoT system in later of Sect. 5. Section 6 formulates mathematically the VNF redundancy allocation problem and proposes a cost-efficient VNFs selection scheme to backup deployment in order to minimize backup costs. Section 7 shows the experiment results to evaluate the efficiency of our solutions. Lastly, the summarization of this paper is presented in Sect. 8.

2 Related Work

Recently, many studies have been investigated to meet service availability and reliability requirements in NFV.

To improve SFC reliability, a simple approach is VNFs placement optimizations. In [3], the authors propose a queue-aware reliable embedding algorithm to improve the reliability of services without reserving the backup resource. An availability-enhanced VNF placing scheme is proposed in [4] to maintain desired end-to-end latency and SFC reliability based on the layered graphs approach. Sun et al. [5] improve SFC reliability and reduce SFC resource consumption by mapping functions onto the substrate network. Although most of the proposed methods that focus purely on original VNFs placement schemes increase the reliability of network services, they are not enough to meet their required high reliability.

Another promising approach to improve the reliability of SFCs is to implement VNF redundancies and combine them with original VNF placements. Some researches on reliable SFCs are presented in [6, 7]. These studies increase services' reliability by creating multiple backups of the least reliable VNF in the chain. A more sufficient scheme to decide on VNF candidates for backup deployment is figured out in [8]. The authors propose the Cost-aware Importance Measure (CIM) associated with the reliability of the physical node (PN) to assess the importance of VNFs. The VNFs with the largest CIM results are selected for redundancy deployment to satisfy the reliability requirement of each SFC with optimal cost-effectiveness. In [9, 16], the authors investigate solutions to ensure the survivability of services avoiding any failure of a PN on the basis of shared backup resources. To increase SFC reliability, the approach in [17] uses a k-shortest path algorithm to deploy primary VNFs and a hybrid routing scheme to deploy backup VNFs. The studies in [6–9, 16, 17] only consider hardware failures, and the reliability of an SFC is determined by the reliability of substrate nodes, which host the VNFs of an SFC.

Some studies on reliable SFCs are investigated on the basis of considering the reliability of VNFs. In [18], the authors study off-site backup VNFs to increase the availability of SFCs to meet the client's requirements. The proposed scheme reduces resource consumption for service providers. In [19], Zhang et al. take into account the heterogeneity of the redundant resource requirements of different VNFs types to minimize the resource consumption of the redundant nodes. A novel online algorithm BCR is proposed in [20] to provide off-site redundancy for reliability-aware wide area service chaining.

Both the functional reliability and underlying hardware reliability are considered to find the optimal solution for the reliability guarantee problem in NFV. A fault-avoidance approach is proposed in [21]. Based on the location of VNFs in the service chain, function type, and operational features, the proposed method assigns priority to important VNFs to ensure that the important VNFs are allocated on highly reliable physical hosts. Thus, the reliability of SFC is improved. In [22], the study maximizes the reliability of network services, however, it replicates VNFs with the same number of redundant replicas resulting in suboptimal deployment costs. Zhai et al. [23] propose a reliability-aware SFC backup method. The proposed method consolidates VNFs to the same server nodes for primary VNF deployment and adds redundant backups. Thus, the SFC reliability is increased, but it cannot take over in case of single-node failures.

Reliability and availability guarantee problems are also investigated specifically for different network architectures. Some researches on VNFs backup implementation in Data Center are presented in [7, 24, 25]. The authors back up the entire SFC which requires a large number of resources and may lead to unnecessary operation expenditure and hardware expense in [25]. In [26], Kaliyammal Thiruvasagam et al. consider both reliability of PNs and VNFs to guarantee the reliability requirements of diverse service requests with the minimal redundant resources in NFV-enabled 5G networks. Because of some special features of resource constraints in the edge layer, such as the limitations of the edge layer resources, these proposed schemes are not sufficient to apply directly to the edge layer.

There are several research works implementing VNFs backup in Edge Cloud Computing. In [10], Dinh and Kim propose a cost-efficient solution that guarantees the availability of IoT services over a fog-core cloud network. The proposed scheme improves the availability of SFCs on the basis of measuring VNFs' improvement potential. Other research works are performed in the Mobile Edge Computing (MEC)–NFV environment to reduce costs and enhance the availability of network services in [11, 12]. These studies protect unreliability VNFs on the basis of considering software or hardware failures separately and ignore the impact of the reliability of physical hardware on SFCs' reliability when multiple VNFs are deployed on the same server, resulting in inaccurate reliability estimates. A backup-enabled embedding problem in NFV is investigated to minimize the resource consumption for large-scale edge computing in [13]. The obtained solution can meet the availability requirements of service chains based on considering the availability of both VNFs and edge servers, but it may not ensure service chains' availability when an edge server goes down. Li et al. focus on reliable VNF service provisioning in MEC by deploying redundant VNF instances to meet the reliability requirements of mobile users in [15]. A similar study is investigated in [14]. In the proposed schemes, the authors consider both reliability of VNFs and the cloudlet at which the VNFs instances are located, however, they assume that each mobile user requests only one VNF instance service per request while NFV services consist of several VNFs in our problem. Sang et al. propose a scheme to efficiently back up VNFs in [27]. Specifically, the proposed scheme finds for each VNF, requested in the edge, a suitable place in the edge and cloud layers to deploy a static backup. Since the availability of SFCs may not be guaranteed yet and it is difficult to predict future failures of VNFs, the paper proposes a dynamic backup scheme while balancing the load of each server to mitigate resource contention without assuming the failures of VNFs. In the paper, the authors ignore the reliability of both physical machines and VNFs, so it is not suitable for some network services that adequately and strictly require high reliability.

In summary, existing redundancy methods protect the unreliable VNFs for end-to-end service based on optimizing the composition and mapping SFCs. However, there are still some problems remaining to be solved such as not being high enough to meet the reliability of network services; not being sufficient to apply directly to the edge layer because of the limitations of the edge layer resources; ignoring the impact of the reliability of physical hardware on SFCs' reliability when multiple VNFs are deployed on the same server, resulting in inaccurate reliability estimates, or cannot taking over single node failures. To get over the shortcomings, in this paper, we first design an efficient reliability-aware embedding mechanism for both original VNFs and corresponding backup VNFs based on considering both hardware and software reliability. On the one hand, the proposed model ensures the tolerance of SFCs that can take over in case of any failure of a single node. On the other hand, it gets a sweet point between increasing the minimal reliability of SFCs overall service demands and the total deployment cost. We then formulate the VNFs redundancy allocation cost minimization problem. The redundancy scheme backs up VNFs in the edge layer to increase the reliability of SFCs to a given requirement on the basis of considering both hardware and software reliability, as well as VNFs' resource consumption.

3 Background

3.1 Reliability Models

In this section, we present a model to estimate the reliability of a network service consisting of several interconnected network functions. In this model, the reliability of network functions is assumed to be independent. Other VNF's reliability definitions and assumptions are also given by ETSI [2].

3.1.1 Reliability of a Single Component

A complex system consists of some constituent components such as a service chain consisting of several VNFs in a given order. Therefore, to evaluate the reliability of a composite component, it is necessary to know the reliability of individual components that make it up. We assume that each PN k has the reliability r_k which can be estimated by mean time between failure, MTBF [2]. The reliability of a PN is independent of other PNs or the load imposed on it. We also assume that a VNF f has the reliability r_f, and the reliability of all VNFs is independent. The reliability of a VNF is also assumed that it is not affected by PN on which it is deployed. So, the reliability of a single VNF f deployed in PN k is calculated as follows:

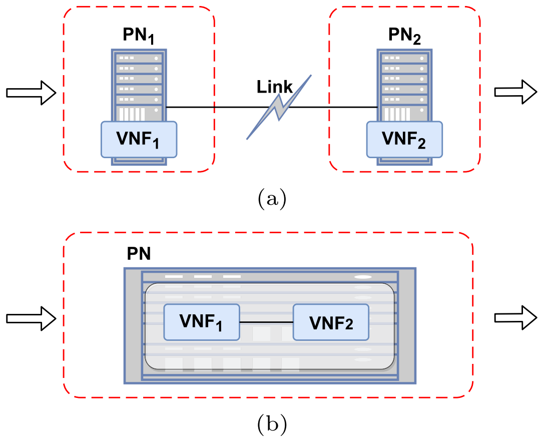
$$R_{\mathrm{VNF}} = r_f \cdot r_k. \tag{1}$$

3.1.2 Reliability of a Composed System

A network service is commonly composed of several VNFs in a given order. The reliability of such SFCs is derived from the individual components that make them up. Therefore, they depend on how their components are combined, in a serial or parallel manner.

In a serial manner, the individual components are connected sequentially, so they are required to be available at the same time to provide services for an SFC. There are two cases of placing VNFs on PNs in a series. In the first case, as shown in Fig. 1a each VNF is located on a different PN. And the second case, both VNFs are located on the same PN as shown in Fig. 1b. To process data

Fig. 1 VNFs in serial ways



traffic from services using all required functions as given in the SFC, both VNF_1 and VNF_2 need to be available simultaneously. Therefore, the reliability of these SFC requests in a sequence is:

$$R_{\text{SFC}} = \prod_{i \in \{1,2\}} (r_{k_i} \cdot r_{f_i}), \tag{2}$$

$$R_{\text{SFC}} = r_k \prod_{i \in \{1,2\}} (r_{f_i}). \tag{3}$$

In the second way, as shown in Fig. 2, the individual components are connected in a parallel manner. If both VNF_{11} and VNF_{12} provide the same function and these VNFs are placed on the same PN as shown in Fig. 2a, the requested service is available when at least one of these VNFs is operable. Thus, the reliability of this SFC request can be estimated as follows:

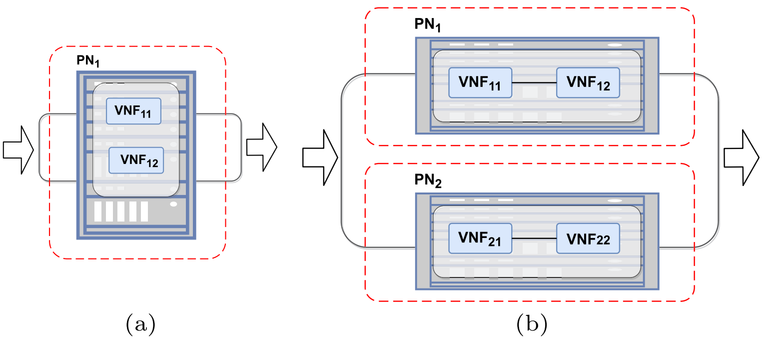


Fig. 2 VNFs in parallel ways

$$R_{SFC} = r_{k_1} \cdot \left(1 - (1 - r_{f_{11}}) \cdot (1 - r_{f_{12}})\right). \tag{4}$$

In the other case, as shown in Fig. 2b, the reliability of a subcomponent consisting of two parallel sub-network chains is calculated as follows:

$$R_{SFC} = 1 - \prod_{i\in\{1,2\}} \left(1 - r_{k_i} \cdot \prod_{j\in\{1,2\}} r_{f_{ij}}\right). \tag{5}$$

3.2 Motivation

To motivate our work, we illustrate an example of how to improve the reliability of a network service by using the location of both original VNFs and their corresponding backups. In our example, we consider an SFC consisting of two VNFs. These VNFs are deployed at the same PN, and they have the same reliability parameter $r_{f1} = r_{f2} = 0.9$. The PN has a reliability parameter $r_k = 0.8$. The requested reliability of this SFC is 0.9. Without any backup, the SFC has the reliability r_{SFC}= 0.648 (Fig. 3a). In the first case (Fig. 3b), we back up two VNFs at two different PNs. Both the backup nodes and the backup VNFs copies have a reliability of 0.9. As a result, the reliability of SFC increases from 0.648 to 0.896. The reliability obtained after redundancy deployment is still smaller than the required reliability. Therefore, it is necessary to deploy additional backups. In the second case (Fig. 3c), we back up two VNFs at the same redundant PN. The reliability of both the backup node and the backup copies of VNFs is set to 0.9. Here, the reliability of the SFC increases from 0.648 to 0.93. So, there is no need for extra backups in the second case. Hence,

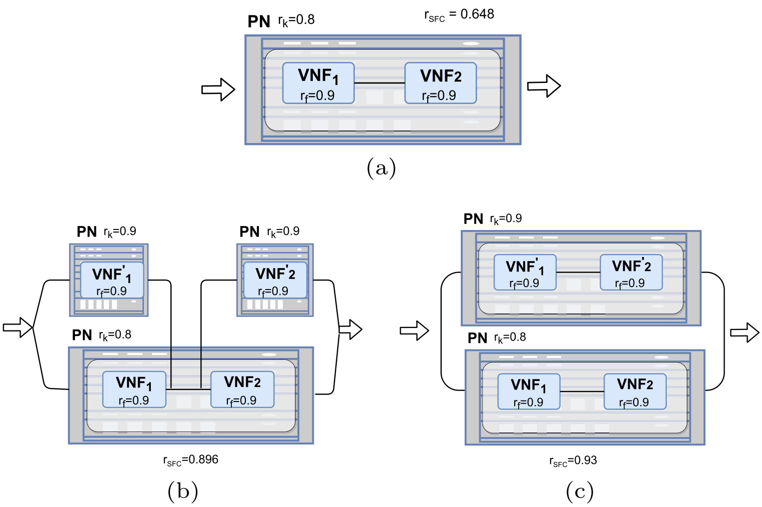


Fig. 3 Example of backup allocation strategies

taking into account both the reliability of VNFs and physical hosts helps to accurately estimate the reliability of network service chains and avoid unnecessary resource consumption.

As we see in the discussed example, the reliability of an SFC can be improved through the proper placement of the original VNFs and the backup copies of VNFs at the PNs.

4 System Description

In this section, we introduce an NIoT system composing of two computing layers: an IoT layer and an edge layer. Edge nodes are NFV infrastructure (NFVI) nodes deployed at the edge layer, and IoT nodes are nodes that are attached to the IoT layer. We present an NIoT system by a directed graph $G = (V, E)$ where $V = V_G \cup V_K$ is a set of nodes in the NIoT system, V_G is a set of IoT devices, V_K is a set of the edge nodes, and $E = \left\{ e_{ij} | i,j \in V \right\}$ is the set of links in the NIoT system. We define c_k is the computing capacity of an edge node k. Let F denote a set of service functions deployed at the edge layer.

In an NIoT system, data are collected from end nodes to IoT gateways. Depending on services requested from customers, the data traffic then is routed to the edge layer. We first consider an efficient reliability-aware placement problem for original VNFs and their corresponding backup VNFs in order to ensure the tolerance of SFCs to edge node failures. For more details, we deploy a backup copy (called full-backup VNFs) for each original VNF (called primary VNFs) in the edge layer to enhance the reliability of SFCs. Then, we optimize the location of primary VNFs and their backups for multiple purposes simultaneously including minimizing the deployment costs and increasing the minimal reliability of SFCs overall service requests with the limitations of resources in the edge layer. Next, we consider a reliability guarantee problem to minimize the total costs of VNFs redundant deployment while maintaining all reliability requirements and resource constraints in the edge layer. For these problems, we do not consider cases that some VNFs of an SFC are backed up at one node and others are backed up at other nodes at the edge layer. The reason is that placing the VNFs of an SFC at different nodes adds both more switchover time to end-to-end delay and extra traffic to the network.

In the optimization problems, in the edge layer, we assume that a VNF f has reliability r_{1f} and a PN k is associated with reliability r_{2k}. Next, we assume that the reliability of all VNFs is independent and it is not affected by PN on which VNFs are located. We assume that a request demand $d_g \in D$ from an IoT gateway g is denoted by $d_g = (b_g, r_{3g}, F_g)$ where b_g is total traffic passing through the IoT gateway g, r_{3g} stands for the reliability requirement, and SFC's F_g is a set of VNFs required by request demand d_g in the edge layer. Let w_f be the number of computing resources that are required for providing function f for one unit of traffic. ζ_k is the deployment unit cost of node k to provide network function for one unit of traffic. We state these problems as follows.

Problem 1 (primary VNFs and full-backup VNFs placement): given $G = (V, E)$ and a set of service requests D, find a mapping solution for primary VNFs and their corresponding backup VNFs that ensures the survivability of SFCs from any failure of a single node to minimize the deployment costs and maximize the minimal reliability of SFCs overall service requests with the resource limitations of the edge layer.

Problem 2 (reliability guarantee problem): given $G = (V, E)$ and a mapping solution for primary VNFs and full-backup VNFs from a set of service requests D, find a cost-effective solution of VNFs backup deployments in the edge layer to meet the high reliability requirements of SFCs while maintaining resource constraints.

5 Optimized Primary VNFs and Full-Backup VNFs Placement Strategy

5.1 Mathematical Formulation

In this section, we present an efficient reliability-aware primary VNFs and full-backup VNFs placement mechanism. In Table 1, we summarize important notations used in this paper.

To ensure the tolerance of SFCs to edge node failures, we deploy a backup copy of each original VNF. To simplify the optimization problem, we also assume that all VNFs of an SFC are located on one PN and all their corresponding backup VNFs are also located on another PN. Due to the resource limitations of the edge layer, the task is to find a mapping solution for primary VNFs and full-backup VNFs to minimize the deployment costs and maximize the minimal reliability of SFCs. The variables are as follows:

- $Y = (y_k^g : g \in V_G, k \in V_K)$ is an original VNFs location vector for all service demands where $y_k^g = 1$ is a binary variable that represents whether node k provides the SFC of service request d_g. If node k provides the SFC of the service request d_g, $y_k^g = 1$, otherwise $y_k^g = 0$.
- $B = (\beta_k^g : g \in V_G, k \in V_K)$ is a full-backup VNFs location vector for all service demands where $\beta_k^g = 1$ is a binary variable that represents whether the SFC of the service request d_g is deployed a backup copy of original SFC on node k in the edge layer, otherwise $\beta_k^g = 0$.

Condition (6) makes sure that all primary VNFs required by service request d_g are satisfied.

$$\sum_{k \in V_K} y_k^g = 1, \ \forall g. \tag{6}$$

To make sure that each primary VNFs required by service request d_g has one backup copy VNF (full-backup VNFs), we use a constraint (7) as follows:

$$\sum_{k \in V_K} \beta_k^g = 1, \ \forall g. \tag{7}$$

Table 1 Summary of notations

Input parameters	
$G = (V, E)$	A directed graph represents an NIoT system. $V = V_G \cup V_K$ is the set of nodes where V_G is the set of IoT devices, and V_K is the set of edge nodes in the NIoT system
D	A set of request demand
F	A set of network functions are provided in the edge layer
w_f	An amount of computing resource is required for providing function f for one unit of traffic
r_{1f}	The reliability of a VNF f
r_{2k}	The reliability of a physical node k
c_k	The computing capacity of a physical node k
d_g	A service request passing an IoT gateway $g \in V_G$
b_g	The total traffic of a service request d_g
r_{3g}	The reliability requirement of the SFC of gateway g
F_g	A set of network functions are required by demand d_g in the edge layer
ζ_k	The deployment cost for each resource unit at node k in the edge layer
Output variables	
Y	An original VNFs location vector for all service demands $Y = (y_k^g : g \in V_G, k \in V_K)$
B	A full-backup VNFs location vector for all service demands $B = (\beta_k^g : g \in V_G, k \in V_K)$
y_k^g	A binary variable that represents whether node k provides the SFC of service request d_g. If node k provides the SFC of the service request d_g, $y_k^g = 1$, otherwise $y_k^g = 0$
β_k^g	A binary variable that represents whether an SFC of service request d_g is deployed a backup copy of original SFC on node k in the edge layer, otherwise $\beta_k^g = 0$
γ_{of}^g	The number of backup copies of original VNF f of the SFC of service request d_g
γ_{bf}^g	The number of backup copies of the full-backup VNF f of the SFC of service request d_g
Others	
a_k	The total resources are required to deploy primary VNFs and full-backup VNFs on PN k
R_g	The reliability of service request d_g
R_{og}	The reliability of the original SFC of service request d_g
R_{bg}	The reliability of the backup SFC of service request d_g
$\overline{R}$	The minimal reliability of all SFCs
Ψ_k	The total resources are required to deploy extra backup copies for the original VNFs and full-backup VNFs on PN k
Q	A continuous variable is used in Algorithm 1 to represent temperature
Q_0	A parameter represents initial temperature
L	A parameter represents the times of the inner while-end loop

To avoid single node failures, the following constraint ensures that the primary VNFs and full-backup VNFs cannot be located on the same PN.

$$\beta_k^g + y_k^g \le 1, \quad \forall g \in V_G, \quad k \in V_K. \tag{8}$$

The capacity requirements to deploy all network functions at a node should not violate the available resource capacity of that node, which can be expressed mathematically as follows:

$$a_k = \sum_g \left(\left(y_k^g + \beta_k^g \right) \cdot b_g \cdot \sum_{f \in F_g} w_f \right), \tag{9}$$

$$a_k \leq c_k, \quad \forall k \in V_K. \tag{10}$$

We use R_g to represent the reliability of service request d_g passing through IoT gateway g. As our above assumptions, each service request from gateway g has an original SFC consisting of primary VNFs and a backup SFC that is chained up of a sequence of full-backup VNFs. Hence, the reliability of service request d_g is calculated as follows:

$$R_g = 1 - (1 - R_{og})(1 - R_{bg}) = R_{og} + R_{bg} - R_{og} \cdot R_{bg}, \tag{11}$$

where R_{og} denotes the reliability of the original SFC, and R_{bg} denotes the reliability of the backup SFC of service request d_g. R_{og} and R_{bg} are calculated according to formulations (12) and (13) as follows:

$$R_{og} = \sum_{k \in V_K} \left(y_k^g \cdot r_{2k} \right) \cdot \prod_{f \in F_g} r_{1f}, \quad \forall g \in V_G, \tag{12}$$

$$R_{bg} = \sum_{k \in V_K} \left(\beta_k^g \cdot r_{2k} \right) \cdot \prod_{f \in F_g} r_{1f}, \quad \forall g \in V_G. \tag{13}$$

The reliability of SFCs is estimated according to Eq. (11), which leads to some of quadratic constraints in this model because of the multiplication of binary variables. Hence, we use a binary variable $z_{ij}^g = y_i^g \cdot \beta_j^g$ to convert (11) into a linear equation:

$$R_g = R_{og} + R_{bg} - \sum_{i,j \in V_K} \left(z_{ij}^g \cdot r_{2i} \cdot r_{2j} \right) \cdot \prod_{f \in F_g} r_{1f}^2, \quad \forall g \in V_G. \tag{14}$$

Now, the relationship between z_{ij}^g, y_i^g and β_j^g becomes:

$$z_{ij}^g \leq y_i^g, \tag{15}$$

$$z_{ij}^g \leq \beta_j^g, \tag{16}$$

$$z_{ij}^g \geq 0, \tag{17}$$

$$z_{ij}^g \geq y_i^g + \beta_j^g - 1, \quad \forall i,j \in V_K, \quad g \in V_G. \tag{18}$$

Our objective is to find the optimal solution that minimizes the deployment costs and maximizes the minimal reliability of all SFCs. The objective function of the problem is as follows:

$$\max \left(\alpha \cdot \bar{R} - \delta \cdot \bar{C}\right), \tag{19}$$

where

$$\bar{R} \leq R_g, \quad \forall g \in V_G, \tag{20}$$

and

$$\bar{C} = \frac{\sum_{k \in V_K} a_k \cdot \zeta_k}{2 \cdot \sum_g \left(b_g \cdot \sum_{f \in F_g} w_f\right) \cdot \max_{k \in V_K}(\zeta_k)}. \tag{21}$$

In detail, the first term of the expression (19) represents the aim of maximizing the minimal reliability of overall SFCs. The second term implies the aim of minimizing the total deployment costs. For more details, the numerator of $\bar{C}$ is the total cost to deploy all primary VNFs and full-backup VNFs in the edge layer. The denominator of $\bar{C}$ evaluates the largest possible value of the total costs to deploy all primary VNFs and full-backup VNFs in the edge layer. It is the product of two factors $2 \cdot \sum_g \left(b_g \cdot \sum_{f \in F_g} w_f\right)$ and $\max_{k \in V_K}(\zeta_k)$. The first factor is the total computing resources required to deploy both original and full-backup VNFs for processing the total traffic in the network. In this equation, we have factor **2** because each demand requests $\sum_g \left(b_g \cdot \sum_{f \in F_g} w_f\right)$ units of computing resources to deploy its original VNFs and the same number of resources to deploy their full-backup VNFs. The second one is the largest value of the deployment cost for a resource unit at edge nodes. Therefore, with the negative sign, the second term of the objective function expression represents the aim of minimizing the total deployment costs. Last, α, δ are the model parameters representing weights to control the importance of the minimal reliability of SFCs overall service requests and the total deployment cost, respectively.

5.2 Approximation Algorithms

In the previous section, we propose a MILP model to find the optimal solution of the joint placement problem of original VNFs and full-backup VNFs in an NIoT system. However, the MILP solvers often fail to solve a large model with hundreds of service requests and edge servers. Hence, in this section, we propose an approximation algorithm for a large-scale NIoT system (presented in Algorithm 1) to minimize the total deployment costs for reliability-aware primary VNFs and full-backup VNFs placement problem.

The primary concept of the algorithm approach is on the basis of the **S**imulated **A**nnealing (SA) with the development of the neighborhood selection where the solution represents the optimal placement solution of primary VNFs and their corresponding backup VNFs in the system. SA is a heuristic approach to find the global optimum for the optimization problem in a set of possible solutions that may contain some local optimums. To overcome local optimal solutions, it may move to a worse scenario with a certain probability. In that way, this algorithm simply and effectively escapes from local optimums.

At first, the algorithm uses GREEDY function (presented in Algorithm 2) to find a feasible placement solution for primary VNFs and their corresponding backups. The algorithm then finds the optimal solution by two loops underlying the neighborhood selection scheme. Specifically, for two loops, we use Q_0 (line 4) and L (line 7) as two parameters to set up the initial temperature and the times of the inner while-end loop, respectively. The algorithm uses NeighborGeneration function (presented in Algorithm 3) to generate a new neighborhood (line 8). Next, based on the new neighborhood, the algorithm estimates the objective function O' according to Eq. (19). Lines 11 and 12 present that the new obtained solution is better than the current one if and only if the value of the objective function O' is not worse and the number of backed up SFCs N do not decrease. The optimization process updates the current solution in the next loop by the neighborhood solution if it is better or with probability $e^{-\Delta O/Q}$ to overcome a local optimum. Finally, the algorithm achieves the optimal primary and full-backup SFC embedding solution.

Algorithm 1 The reliability-aware primary VNFs and full-backup VNFs embedding approximate algorithm

Input: G, D, Q_0, L

Output:

The optimal solution for primary and full backup placement scheme Y, B

$(Y, B) \leftarrow$ GREEDY(G, D)
$(N, O) \leftarrow$ CACULATE(Y, B)
$N_{best} \leftarrow N, O_{best} \leftarrow O$
$Q \leftarrow Q_o$
while $Q > 1$ **do**
  $i \leftarrow 0$
  while $i < L$ **do**
    $\left(Y^{'}, B^{'}\right) \leftarrow$ NEIGHBORGENERATION(Y, B)
    $\left(N^{'}, O^{'}\right) \leftarrow$ CACULATE$\left(Y^{'}, B^{'}\right)$
    $\Delta N \leftarrow N^{'} - N$, $\Delta O \leftarrow O^{'} - O$,
    if $\Delta N \geq 0$ **and** $\Delta O \geq 0$ **then**
      $(Y, B) \leftarrow \left(Y^{'}, B^{'}\right)$
      if $O^{'} > O_{best}$ **then**
        $N_{best} \leftarrow N^{'}, O_{best} \leftarrow O^{'}$
      end if
    else
      if random(0,1) $< e^{-\Delta O/Q}$ **then**
        $(Y, B) \leftarrow \left(Y^{'}, B^{'}\right)$
      end if
    end if
    $i \leftarrow i + 1$
  end while
  $Q \leftarrow ReduceTemperature(Q)$
end while

Algorithm 2 The greedy procedure for the reliability-aware primary VNFs embedding mechanism

Input: G, D

Output: the primary and full backup placement scheme Y, B

function GREEDY(G, D)
    Sort the traffic requests in set of request D with respect to the requested reliability of SFCs and the total traffic value in descending order
    $V_K^{'} \leftarrow$Sort V_K with respect to the r_{2k} in descending order and ζ_k in ascending order
    for $d_g \in D$ **do**
        $\widetilde{V}_K \leftarrow V_K^{'}$
        do
            $k \leftarrow$ the first node of $\widetilde{V}_K$
            $\widetilde{V}_K \leftarrow \widetilde{V}_K \backslash k$
            if $\left(a_k + b_g . \sum_{f \in F_g} w_f \right) \leq c_k$ **then**
                $y_k^g = 1$
                $a_k \leftarrow a_k + b_g . \sum_{f \in F_g} w_f$
            end if
        while $\sum_{k \in V_K} y_k^g = 0$ and $\widetilde{V}_K \neq \emptyset$
    end for
    for $d_g \in D$ **do**
        $\widetilde{V}_K \leftarrow V_K^{'}$
        do
            $k \leftarrow$ the first node of $\widetilde{V}_K$
            $\widetilde{V}_K \leftarrow \widetilde{V}_K \backslash k$
            if $\left(a_k + b_g . \sum_{f \in F_g} w_f \right) \leq c_k$ and $y_k^g \neq 1$ **then**
                $\beta_k^g = 1$
                $a_k \leftarrow a_k + b_g . \sum_{f \in F_g} w_f$
            end if
        while $\sum_{k \in V_K} \beta_k^g = 0$ and $\widetilde{V}_K \neq \emptyset$
    end for
end function

Algorithm 3 Algorithm for neighborhood generation of the reliability-aware primary VNFs embedding mechanism

function NeighborGeneration(Y,B)
  $\epsilon \leftarrow$ a random number in (0,1)
  $g^{'} \leftarrow$ a random number in $[1, |V_G|]$
  $k^{'} \leftarrow$ a random number in $[1, |V_K|]$
  if $\epsilon < \eta$ **then**
    if $y_{k'}^{g'} \neq 1$ and $\left(a_{k'} + b_{g'} . \sum_{f \in F_{g'}} w_f \right) \leq c_{k'}$ **then**
      $\beta_{k'}^{'g'} \leftarrow 1$
      $\beta_{k}^{'g'} \leftarrow 0, \forall k \neq k^{'}$
      $y_{k}^{'g} \leftarrow y_{k}^{g}$

    end if
  else
    if $\beta_{k'}^{g'} \neq 1$ and $\left(a_{k'} + b_{g'} . \sum_{f \in F_{g'}} w_f \right) \leq c_{k'}$ **then**
      $y_{k'}^{'g'} \leftarrow 1$
      $y_{k}^{'g'} \leftarrow 0, \forall k \neq k^{'}$
      $\beta_{k}^{'g} \leftarrow \beta_{k}^{g}$

    end if
  end if
  return $\left(Y^{'}, B^{'} \right)$
end function

The GREEDY function is designed to find a feasible solution for primary VNFs and full-backup VNFs placement with some of the following purposes. For the first purpose, all original VNFs and their corresponding backup VNFs of the GREEDY solution are placed in different servers against any failure of a single node. The second one is that the minimal reliability of SFCs overall service requests is as large as possible. The main idea of the algorithm is to map SFCs with the largest reliability requirement for highly reliable edge servers. The details of the GREEDY function are presented in Algorithm 2. First, all service requests from gateways are sorted with respect to their reliability requirements and the amount of their traffic volume in descending order (line 2). Next, all edge servers are sorted firstly in descending manner according to their reliability and secondly in ascending order with respect to their deployment unit cost (line 3). The set of edge servers after sorting is called V_K'. Then, with each SFC the algorithm finds a proper edge node to deploy original VNFs of that SFC (lines 4 to 14) and another node for their corresponding backup deployment (lines 15 to 25). In detail, the algorithm finds the first node of V_K' which has enough resources to deploy all VNFs of an SFC (the consideration is presented at line 9 for original VNFs and line 20 for backup instances). In the **if** statement at line 20, the second conditional expression ensures that the node holding original VNFs of an SFC does not deploy the corresponding backup VNFs instances according to the reliability requirement in case of hardware failures.

The details of the NeighborGeneration function for neighborhood generation of primary VNFs and full-backup VNFs placement are presented in Algorithm 3. Based on a feasible solution, the NeighborGeneration function generates another one with some following steps. For the first step, ϵ gets its value as a random number uniformly distributed between 0 and 1. Then, the algorithm also randomly chooses a gateway g' (line 2) and an edge node k' (line 3). Next step, all full-backup VNFs instances of SFC from gateway g' are considered to move to edge node k' if $\epsilon < \eta$ (lines 5 to 11), where η is a given parameter with its value on the interval (0, 1). Otherwise, all original VNFs instances of SFC from gateway g' are considered to move to edge node k' (lines 12 to 19). A new feasible solution is created only if node k' has enough available resources to deploy VNFs of SFC from gateway g' and node k' is not holding either original VNFs instances (line 6) or full-backup VNFs instances (line 13).

After all of the primary VNFs and backup VNFs are deployed, these service requests are protected from the failures of a single node in the edge layer, however, the reliability of SFCs may not meet each request's reliability requirements. Therefore, we propose a redundancy allocation scheme to improve the reliability of SFCs to meet high reliability requirement of each request in the next section.

6 Optimized VNF Redundancy Allocation Strategy

In this section, we define a VNF redundancy allocation cost minimization problem to achieve the high requested reliability of SFCs based on considering both hardware and software failures, and VNFs' resource consumptions. We first give out the mathematical formulation of the problem. We then propose an algorithm to find a feasible solution of the redundancy problem.

6.1 Mathematical Formulation

Section 5 presents an approach for the joint placement optimization problem of primary VNFs and full-backup VNFs in the edge layer of NIoT systems. The found solutions allow against any failure of a single edge node and enhance the reliability of each service demand. However, the obtained reliability of SFCs is normally not enough to meet their request, so more VNFs redundancy deployments are thus required. With the found solutions, we have a set of VNFs with different reliability and each of them has one backup instance. The original VNFs and their full-backup are deployed at edge nodes with different reliability. The next question is which VNF instances are more effective-cost to deploy backups. We now present a mathematical formulation model to achieve the optimal solution for the VNFs backup deployment.

The goal is to enhance the reliability of each service demand to meet their request with the least amount of costs for redundant deployment.

We assume that b_g is the total traffic of service request d_g passing IoT gateway g in V_G: $d_g = (b_g, r_{3g}, F_g)$ where the SFC of F_g consisting of requested VNFs deployed in the network. We further assume that any single VNF failure will break down the whole service. We also assume that all the backup nodes and backup VNF copies have the same value of reliability with their corresponding original instances. Let $\bar{y}_v^g$ and $\bar{\beta}_v^g$ represent whether node $v \in V_K$ provides resources for the primary SFC and backup copy SFC of service request d_g. They are also extracted from solutions of the reliability-aware primary VNFs and full-backup VNFs placement problem that is aforementioned. They are also extracted from solutions of the reliability-aware primary VNFs and full-backup VNFs placement problem that is aforementioned. The obtained reliability of SFCs may not meet their request, so more backup VNFs are thus required besides the primary VNFs and full-backup VNFs. We will deploy parallel several extra backup copies in the same PN for both primary VNFs and full-backup VNFs. The variables are as follows:

- γ_{of}^g is the number of backup copies for primary function f from the SFC of service request d_g.
- γ_{bf}^g is the number of backup copies for full-backup function f of the SFC of service request d_g.

The condition is given by:

$$\gamma_{of}^g \geq 0, \quad \forall g \in V_G, \quad f \in F_g, \tag{22}$$

$$\gamma_{bf}^g \geq 0, \quad \forall g \in V_G, \quad f \in F_g. \tag{23}$$

We use Ψ_k to denote the amount of resource consumption for deploying backups on an edge node k. The formulation is as follows:

$$\Psi_k = \sum_g b_g \cdot \left(\overline{\beta}_k^g \cdot \sum_{f \in F_g} \gamma_{bf}^g \cdot w_f + \bar{y}_k^g \cdot \sum_{f \in F_g} \gamma_{of}^g \cdot w_f \right). \tag{24}$$

Our objective is to find the solution that minimizes the backup costs so that the objective function is formulated as follows:

$$\min \sum_k \left(\zeta_k \cdot \Psi_k \right), \tag{25}$$

where ζ_k is the deployment cost for each resource unit at node k in the edge layer.

Since primary VNFs and their full-backup instances are running, which consumes the amount of resource a_k according to Eq. (9), we formulate a constraint (26) to ensure that the overall resources of VNFs at a node k cannot exceed the total computing resources of that node c_k.

$$\Psi_k \leq c_k - a_k, \quad \forall k \in V_K. \tag{26}$$

In an NFV environment, an SFC is considered to be available at a given time if all the requested functions can function normally. In this problem, we deploy γ_{of}^g backup copies for primary function f of the SFC of service request d_g. So that, the particular reliability for the original SFC is calculated, concerning only the reliability of software as Eq. (28). Similarly, each full-backup instance of function f of the SFC of service request d_g is deployed γ_{bf}^g backup copies, and the particular reliability for the backup SFC is calculated, concerning only the reliability of software as Eq. (29). The original SFC and its corresponding backup SFC are held at edge nodes with different reliability [$\sum_{k\in V_K}(\overline{\beta}_k^g \cdot r_{2k})$ and $\sum_{k\in V_K}(\overline{y}_k^g \cdot r_{2k})$, respectively]. Therefore, the end-to-end reliability of service requests d_g is calculated based on the particular reliability of the original SFC, the backup SFC, and corresponding server nodes. Then, the end-to-end reliability of service requests d_g after redundancy can be obtained as:

$$R'_g = 1 - \left(1 - \sum_{k\in V_K}\left(\overline{\beta}_k^g \cdot r_{2k} \cdot \Omega_b^g\right)\right)\left(1 - \sum_{k\in V_K}\left(\overline{y}_k^g \cdot r_{2k} \cdot \Omega_o^g\right)\right), \tag{27}$$

where

$$\Omega_o^g = \prod_{f\in F_g}\left(1 - (1 - r_{1f})^{\gamma_{of}^g+1}\right), \tag{28}$$

and

$$\Omega_b^g = \prod_{f\in F_g}\left(1 - (1 - r_{1f})^{\gamma_{bf}^g+1}\right). \tag{29}$$

An SFC request is considered blocked if it cannot map any VNF or its reliability cannot meet the client's request. Therefore, we need to back up some VNFs to achieve SFC reliability requirements. The constraint is as follows:

$$R_g{}' \geq r_{3g}, \forall g. \tag{30}$$

6.2 Algorithms

In the previous section, we present a mathematical formulation to find the optimal solution of the redundancy problem in an NIoT system. We now present an algorithm (presented in Algorithm 4) to find a feasible solution for the problem.

The primary idea of the algorithm is based on an effective-cost VNFs selection scheme to deploy VNFs backup copies. In the beginning, the algorithm will loop over all service requests. If the reliability of an SFC meets its reliability requirement, then backups are not required (lines 3 and 4). Otherwise, the algorithm estimates the potential of VNFs of SFCs including a primary SFC and a full-backup SFC of that service requests as presented in lines 6 to 10. Next, all VNFs of that service request are sorted according to their fitness in a descending manner. Line 12 implies that the algorithm will circularly choose VNFs to deploy a VNF backup copy. A VNF backup instance deployment will be performed if it is enough resources at the edge node for the implementation of that VNF. By that way, the reliability of that SFC improves. The loop of backup VNFs deployment will stop until the reliability of an SFC meets its reliability requirement.

Algorithm 4 The reliability requirement guaranteeing procedure

Input: G, D, the primary and full-backup placement scheme
Output: The additional redundancy plan for VNFs

1: **for** $d_g \in D$ **do**
2: Calculate $R_g^{'}$ according to Eq.(11)
3: **if** $R_g^{'} \geq r_{3g}$ **then**
4: Redundant backup is not required
5: **else**
6: **for** $f \in F_g$ **do**
7: $\gamma_{of}^{g} \leftarrow 0$, $\gamma_{bf}^{g} \leftarrow 0$
8: $CRM_{\text{of}} \leftarrow CRM_{\text{f}}(\bar{y}_k^g)$
9: $CRM_{\text{bf}} \leftarrow CRM_{\text{f}}(\overline{\beta}_k^g)$
10: **end for**
11: Sort the VNFs of the primary and backup SFC with respect to appropriate CRM_f value in ascending order
12: Start assigning backups VNFs in the sorted VNFs list from the least CRM_f value to the highest CRM_f value in a circular manner
13: **do**
14: $\gamma_{of}^{g} \leftarrow \gamma_{of}^{g} + 1$ if f is a VNF of primary SFC
15: $\gamma_{bf}^{g} \leftarrow \gamma_{bf}^{g} + 1$ if f is a VNF of backup SFC
16: Recalculate $R_g^{'}$ according to Eq.(27)
17: **while** $R_g^{'} < r_{3g}$ and $\Psi_k \leq c_k - a_k, \forall k \in V_K$
18: **end if**
19: **end for**

We now present in more detail the proper VNFs selection scheme to deploy backup instances to minimize redundancy costs. The VNFs selection scheme considers both deployment cost for backing up VNFs and the reliability of VNFs and PNs. In detail, we propose the **C**ost–Reliability **R**elative **M**easure (CRM) measure to evaluate the potential of VNFs for redundant deployment to increase the reliability of SFCs with maximum cost-efficiency. We define the CRM value of a VNF f to evaluate the potential of VNFs as follows:

$$CRM_{\text{f}}\left(\theta_k^g\right) = \frac{r_{1f} \cdot w_f \cdot \sum_{k \in V_K} \theta_k^g \cdot \zeta_k}{\sum_{k \in V_K} \theta_k^g \cdot r_{2k}}, \tag{31}$$

where θ_k^g is a parameter that indicates whether node k provides function f for service request from gateway d_g. The numerator of formulation (31) implies that we prefer to choose which VNF with smaller reliability and cheaper resource consumption. As assumption at Sect. 5, each service request from gateway g has an original SFC that chained up of a sequence of requested VNFs (primary VNFs) besides a backup SFC that consisted of full-backup VNFs. Primary VNFs and their corresponding backups are deployed at different edge nodes with different reliability and deployment costs. Therefore, the denominator of formulation (31) presents that which VNF deployed at a more reliable node is prioritized for backing up.

7 Evaluation

In this section, we present an assessment of our optimization model and proposed algorithms for the joint optimization placement problem of primary VNFs and full-backup VNFs and the VNF redundancy allocation cost minimization problem in the Edge Computing. We start with a summary of various evaluation scenarios, and several parameter settings for the experiments. We then evaluate the performance of our proposed solutions. For the joint optimization placement problem of primary VNFs and full-backup VNFs, we evaluate the performance of our proposed solutions in terms of several major performance metrics, including (1) the total of deployment costs that composed of deployment costs for primary VNFs and redundancy costs for their corresponding backup VNFs; (2) the minimal reliability of SFCs overall service requests; (3) the number of accepted request demands; and (4) the execution time. For the VNF redundancy allocation cost minimization problem, we evaluate the effectiveness of our proposed mechanism to choose proper VNFs for backing up.

7.1 Simulation Setup

Throughout this experiment evaluations, we consider three network instances as represented in Table 2, namely: small, medium, and large networks. In detail, the small network is composed of 20 servers in the edge layer and supports 30 service requests. The medium network is composed of 40 edge servers supporting 10 to 150 service requests. The large network is composed of 100 servers in the edge layer serving 50 to 650 request demands. Edge servers are randomly connected to several other edge nodes and IoT nodes are connected to several edge servers randomly too. For example, the network topology of a smaller network consisting of 5 edge servers

Table 2 Scenarios

Scenarios	Number of edge servers	Max number of VNFs	Min number of service requests	Max number of service requests
Small network	20	6	5	30
Medium network	40	6	10	150
Large network	100	6	50	650

and 5 IoT nodes is presented in Fig. 4. Each PN in the edge layer has a capacity of 100,000 units that offer different resources (such as CPU, memory, etc.) to instantiate VNFs and process data. The unit cost to process a data traffic unit at a server in the edge layer is uniformly distributed from 1 to 5. In addition, their reliability is set in a uniformly distributed manner from 0.9 to 0.96.

Traffic data of each service request needs to be processed with a set of network functions represented by an SFC that is randomly chained up by 1 to 6 VNFs in the edge layer. The computing resource that is required to instantiate VNFs for processing one data traffic unit is randomly generated from 1 to 5. The reliability of a VNF is randomly generated from 0.95 to 0.99.

The total data traffic that needs to be processed of a service request is a random number uniformly distributed from 100 to 1000. The reliability requirement of each SFC request is generated randomly in uniform distributed between 0.9 and 0.999. In the reliability-aware mechanism, we choose $\alpha = 80$ and $\delta = 1$ as we give higher priority for increasing the minimal reliability overall SFCs.

We carried out the evaluation in an X64-based PC with a two-core 2.6 GHz Intel Core i7-6600 processor and 16 GB memory. Performance evaluations of our proposed solution are performed in terms of the aforementioned major metrics. All numerical results are computed as the average value in 50 runs. With a confidence level is 95%, confidence intervals are very small to be worth adding to the figures. For specifically, confidence intervals are approximately 10^{-6} with minimal reliability and from 3000 to 10,000 with the total deployment costs.

7.2 Performance Evaluation of Reliability-Aware Primary VNFs and Full-Backup VNFs Embedding Mechanism

In this section, we assess the efficiency of our proposed solutions for the reliability-aware placement problem of primary VNFs and their corresponding backup VNFs.

We first analyze the efficiency of our approach in comparing the results produced by the **S**imulated **An**nealing inspired heuristic algorithm (called SAN) and that of a greedy algorithm (called GREEDY in Algorithm 2) with both the medium and large networks. The evaluation results show that SAN outperforms GREEDY in terms of two metrics including the total of deployment costs, the minimal reliability of overall request demands while ensuring the same number of accepted request demands. It can be seen from Figs. 5 and 6 that the total deployment costs obtained by SAN are much smaller than that of GREEDY. Specifically, Fig. 5b shows that

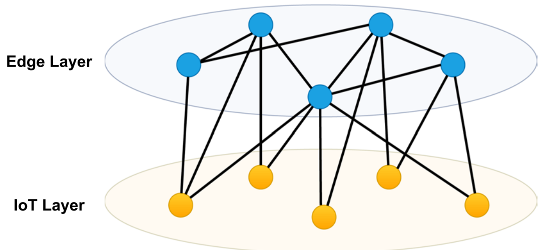
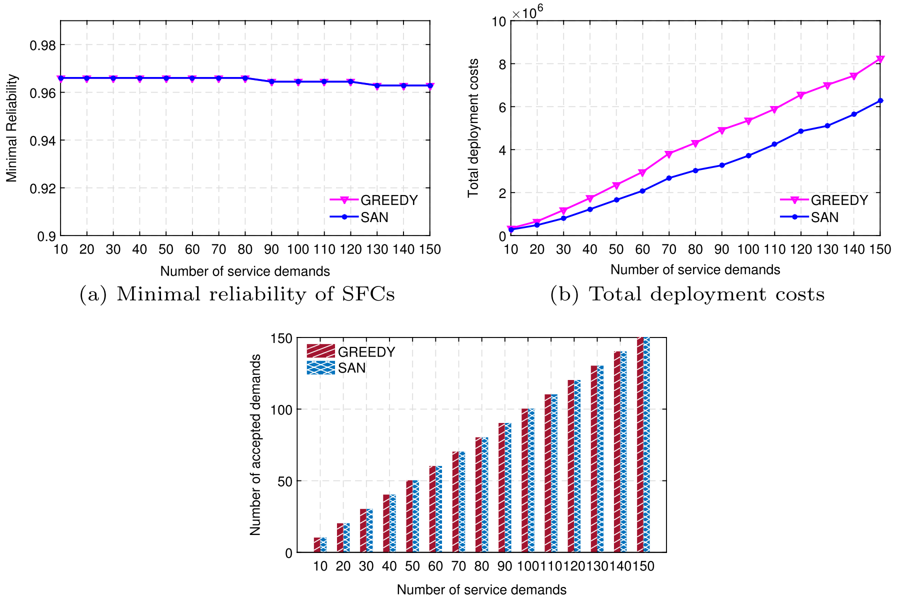


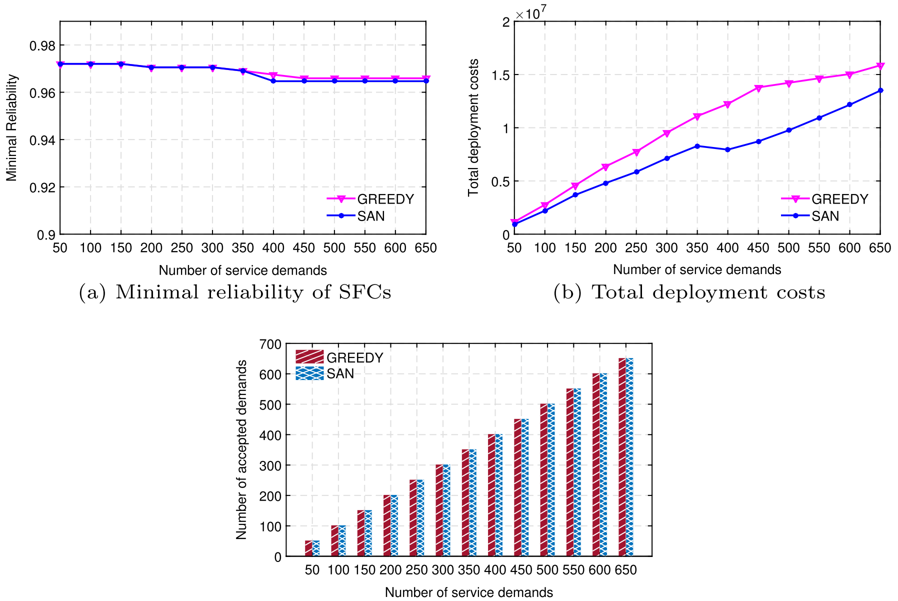
Fig. 4 A network topology example of a smaller network consisting of 5 edge servers and 5 IoT nodes with randomly generated links between edge nodes and edge nodes, or edge nodes and IoT nodes



(a) Minimal reliability of SFCs (b) Total deployment costs

(c) Number of accepted demands

Fig. 5 Comparison between the SAN approximate solution and the GREEDY algorithm for the reliability-aware primary VNFs and full-backup VNFs embedding mechanism with the medium network



(a) Minimal reliability of SFCs (b) Total deployment costs

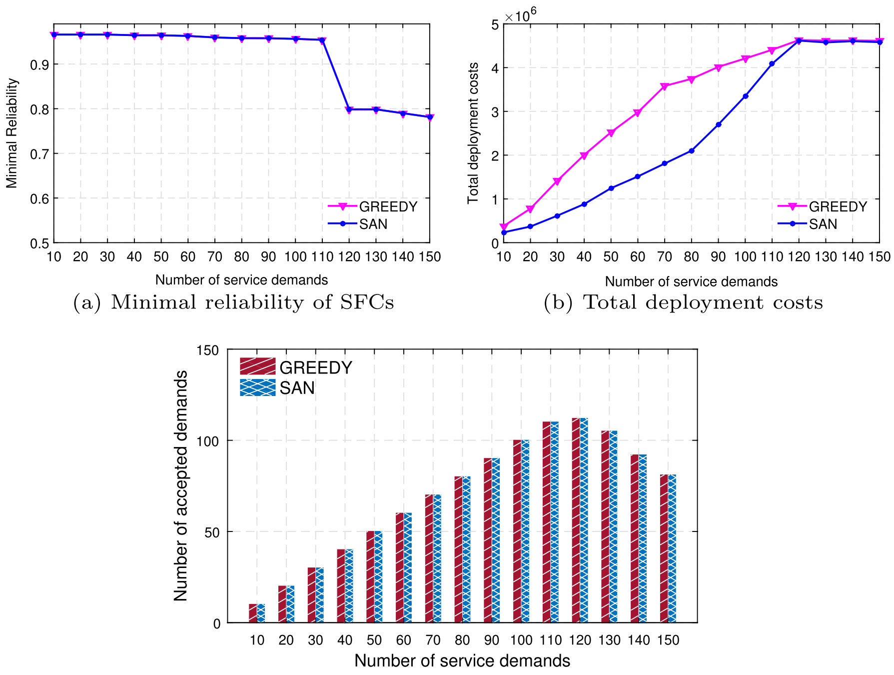
(c) Number of accepted demands

Fig. 6 Comparison between the SAN approximate solution and the GREEDY algorithm for the reliability-aware primary VNFs and full-backup VNFs embedding mechanism with the large network

SAN can save approximately from 18 to 30% of the total deployment cost in comparison with GREEDY for the medium network. In addition, with the large network, SAN requires less approximately from 15 to 37% of the total deployment cost than GREEDY as depicted in Fig. 6b.

To figure out the efficiency of our approach in the case of very limited edge resources, we perform an experiment with the reduction of the capacity of each edge node from 100,000 to 40,000 units in the medium network. That means that for a large number of service requests both GREEDY and SAN may not be able to satisfy all demands as presented in Fig. 7c. It can be seen from Fig. 7b that the total deployment costs obtained by SAN are much smaller than that of GREEDY when supporting the same number of service requests. However, this gap tends to be narrow when the number of service requests increases. The reason is that the resources in the edge layer are not enough to support all the required services, so it is difficult to find a more efficient solution.

We then compare the results obtained by SAN and the optimal results produced by CPLEX (called CPLEX) to solve the MILP models [28] for evaluating the performance of our proposed algorithms with the small network. As shown in Fig. 8, the experiment results show that SAN is effective to find an approximate solution

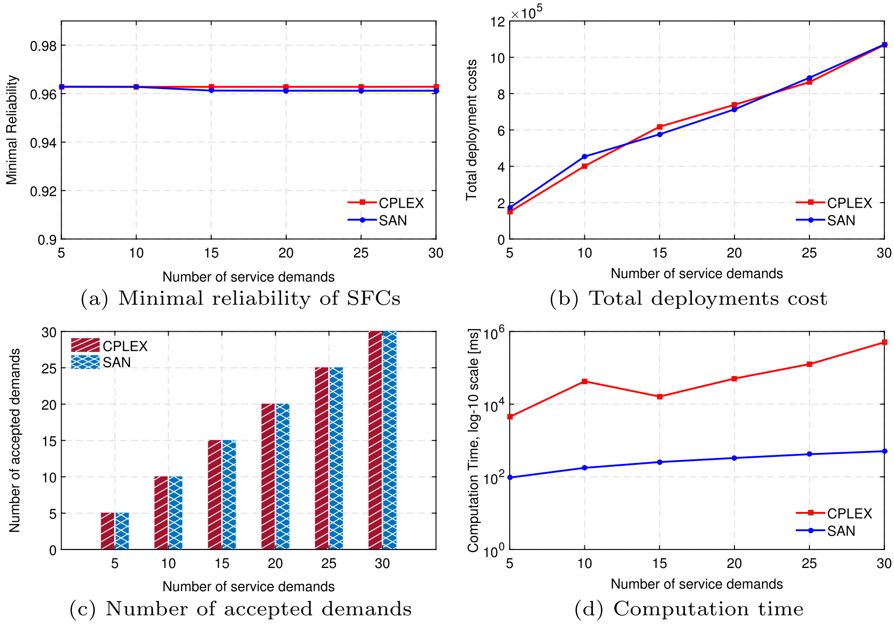


(a) Minimal reliability of SFCs

(b) Total deployment costs

(c) Number of accepted demands

Fig. 7 Comparison between the SAN approximate solution and the GREEDY algorithm for the reliability-aware primary VNFs and full-backup VNFs embedding mechanism with the reconfigured medium network that has the reduction of the computing capacity of edge nodes



(a) Minimal reliability of SFCs (b) Total deployments cost

(c) Number of accepted demands (d) Computation time

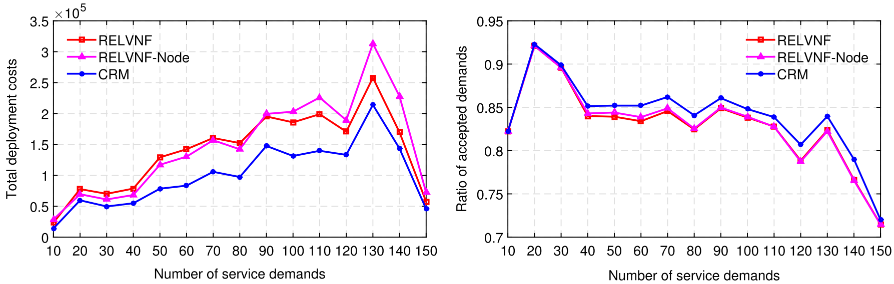
Fig. 8 Comparison between the optimal solution and approximate solution for the reliability-aware primary VNFs and full-backup VNFs embedding mechanism with the small network

for the reliability-aware primary VNFs and full-backup placement problem. Specifically, SAN can satisfy all service demands in order to support full backup against a single error at nodes as same as the optimal solution. In addition, it can be seen from Fig. 8a, b that the minimal reliability overall service demands and the total deployment cost obtained by SAN are very close to the optimal solution. Moreover, SAN outperforms CPLEX in computational time as depicted in Fig. 8d.

To summarize, our proposed algorithm is effective in minimizing deployment costs while guaranteeing the minimal reliability of all request demands for the reliability-aware primary VNFs and full-backup VNFs placement problem.

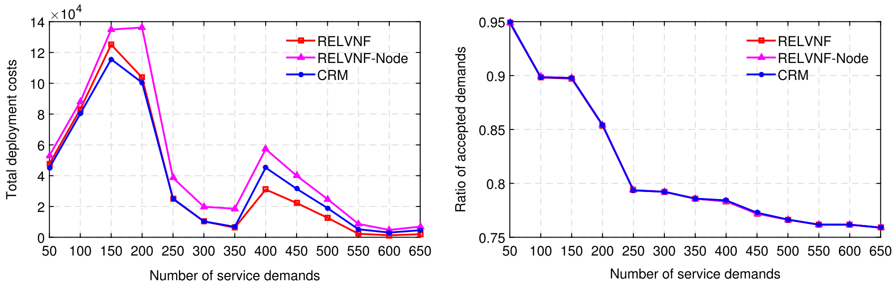
7.3 Performance Evaluation of the VNF Redundancy Allocation Cost Minimization Problem

In the performance evaluation of our proposed solution for the VNF redundancy allocation problem, we compare the experimental results that are obtained by three VNFs selection schemes, including CRM, RELVNF, and RELVNF-Node. These VNFs selection schemes are used to select which more proper VNFs to deploy their backup copies in Algorithm 4. For specifically, CRM uses Eq. (31) for selecting a proper VNF as presented in Algorithm 4. The RELVNF scheme is based on the idea of GREP [6] that prefers to select less reliable VNFs to perform backups. To implement RELVNF, we use Algorithm 4 by removing lines 8–9 and replacing the CRM_f value with the VNF reliability r_{1f} in lines 11–12.



(a) Backup costs (b) Ratio of satisfied demands

Fig. 9 Comparison of the efficiencies of different mechanisms to choose VNFs for backing up with the medium network



(a) Backup costs (b) Ratio of satisfied demands

Fig. 10 Comparison of the efficiencies of different mechanisms to choose VNFs for backing up with the large network

For the RELVNF-Node scheme, we also use Algorithm 4 by removing lines 8–9 and replacing the CRM_f value in lines 11–12 with the multiplication of the VNF reliability r_{1f} and the reliability r_{2k} of PN on which this VNF is implemented. To figure out the effectiveness of the VNF redundancy algorithm under different VNFs selection schemes we compare their backup costs and the ratio of accepted demands, respectively. An SFC request is accepted if and only if the value of its achieved reliability is equal or greater than its requested reliability value.

As can be seen in Figs. 9 and 10, CRM outperforms RELVNF and RELVNF-Node in both the backup cost and ratio of accepted demands with both the medium network and large network. Specifically, Fig. 9a shows that the backup cost of our algorithm using CRM is the smallest in three of VNFs selection schemes with the medium network. For more details, CRM can save from 15 to 50% in terms of backup cost compared to RELVNF-Node and from 16 to 40% in compared to RELVNF. Moreover, with a smaller backup cost, CRM still obtains a larger number of service requests that are backed up to meet high requested reliability in comparison to RELVNF and RELVNF-Node as shown in Fig. 9b. With the large network, Fig. 10b depicts that CRM is better a little bit than RELVNF and RELVNF-Node in terms of the ratio of accepted demands. In Fig. 9a, we can observe that CRM almost requires less cost for VNFs backup deployment than other VNFs selection schemes, however, it is not true in some cases involving a large number of service demands. It is understandable because improving the reliability of SFCs to meet their high requested reliability needs to deployment more VNFs backups. Much more VNFs backups lead to more resource consumptions and more backup costs.

Specially, we can observe that there are two peaks in backup costs over the varying service demands as presented in Fig. 10a. It can be explainable because of the following reason. In this paper, we study a reliability optimization method for given reliability requirements of SFCs. The method is in two steps: the first step finds a map for primary VNFs and their backup VNFs to PNs, and the following step adds more redundant VNFs to meet the requests to the SFCs. In this case, Figs. 9a and 10a only compares the cost of implementing additional redundancy in step 2 to achieve the reliability requirements of each SFCs. Here we want to compare the effectiveness of three VNFs selection strategies for implementing extra redundancy, so the comparison takes place for each number of service requests (each point). In Fig. 10a, when comparing different points (different numbers of service requests), the cost of redundancy deployment is higher while handling a smaller number of service requests at some points, for example at points of 150, 200 requests, and 300, 600 service requests. There are two possible reasons. First, with the same number of resources in the network, the number of resources used to deploy primary VNFs and backup VNFs in step 1 increases in proportion to the number of service requests (Figs. 5b, 6b). That leads to a diminishing number of resources available to deploy additional redundancy in step 2. Therefore, in some cases there will not be enough resources to deploy additional redundancy, resulting in only a few SFCs being provisioned to meet the given reliability requirements. In these cases, the acceptance rate of service requests will be low and the cost of redundancy implementation will be low as seen in Fig. 10b for service request numbers of 250 or more. Second, because of the different number of service requests, the solution of the main VNFs deployment location and the full-backup in step 1 is very different, so it is difficult to compare the additional backup deployment cost between the points with the different number of service requests. For example, at the point of 400 service requests, it is possible that in step 1, many services have not met the requirements for reliability, so it is necessary to deploy many additional backups, leading to high deployment costs.

To sum up, the approach considering both resource consumption of VNFs implementations and the reliability of PNs and VNFs themselves is an effective approach to minimizing the total backup cost for the VNF redundancy allocation problem.

8 Conclusion

The new IoT applications and services provisioned in NFV-enabled IoT systems in the Edge Computing brings about new challenges for high reliability guarantees. Some existing redundancy schemes are proposed to improve the reliability of services, however, they are limited to ensure adequately the survival and high reliability of NFV, especially for network service deployed in the Edge Computing due to the limitations of edge resources. Hence, in this paper, we propose the joint optimization placement problem of primary VNFs and their corresponding backup VNFs in the edge layer of an NFV-enabled IoT system in Edge Computing to avoid single node failures while reducing the overall costs and increasing the minimal reliability of SFCs overall service requests. We also propose a cost-efficient VNFs redundancy scheme that requires fewer backup costs while maintaining high request acceptance ratio. From our experiments, the proposed redundancy mechanism can reduce the backup cost up to 30–40% while maintaining the same ratio of the satisfied service requests. Our future work will investigate shared backup resources for further cost-effective allocation. Another important direction might be the SFCs placement strategies in multiple edge data-centers.

Author Contributions Conceptualization and methodology, TTLN and TMP; formal analysis, investigation, and resources, TTLN and TMP; writing-original draft preparation, TTLN; writing-review and editing, TTLN, LMP and TMP; supervision, TMP; all authors have read and agreed to the published version of the manuscript.

Funding This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under Grant Number 102.02-2020.13.

Data Availability All results are included in this published article; the results raw output is available from the corresponding author on reasonable request.Consent for Publication Not applicable.

Declaration

Conflict of interest There are no competing interests.

Ethical Approval Not applicable.

Informed Consent Not applicable.

References

1. Yi, B., Wang, X., Li, K., Das, S., Huang, M.: A comprehensive survey of network function virtualization. Comput. Netw. **133**, 212–262 (2018)
2. ETSI: V1.1.1., ETSI GS NFV-REL: Network Functions Virtualisation (NFV), Reliability, Report on Models and Features for End-To-End Reliability. Technical Report. ETSI (2016)
3. Tang, L., Zhao, G., Wang, C., Zhao, P., Chen, Q.: Queue-aware reliable embedding algorithm for 5G network slicing. Comput. Netw. **146**, 138–150 (2018)
4. Xu, Y., Kafle, V.P.: An availability-enhanced service function chain placement scheme in network function virtualization. J. Sens. Actuator Netw. **8**(2), 34 (2019)

5. Sun, J., Zhu, G., Sun, G., Liao, D., Li, Y., Sangaiah, A.K., Ramachandran, M., Chang, V.: A reliability-aware approach for resource efficient virtual network function deployment. IEEE Access **6**, 18238–18250 (2018)
6. Fan, J., Ye, Z., Guan, C., Gao, X., Ren, K., Qiao, C.: GREP: guaranteeing reliability with enhanced protection in NFV. In: Proceedings of the 2015 ACM SIGCOMM Workshop on Hot Topics in Middleboxes and Network Function Virtualization. HotMiddlebox '15, 2015, pp. 13–18. Association for Computing Machinery, New York (2015)
7. Qu, L., Assi, C., Shaban, K., Khabbaz, M.: Reliability-aware service provisioning in NFV-enabled enterprise datacenter networks. In: 2016 12th International Conference on Network and Service Management (CNSM), 2016, pp. 153–159 (2016)
8. Ding, W., Yu, H., Luo, S.: Enhancing the reliability of services in NFV with the cost-efficient redundancy scheme. In: 2017 IEEE International Conference on Communications (ICC), 2017, pp. 1–6 (2017)
9. Aidi, S., Zhani, M.F., Elkhatib, Y.: On improving service chains survivability through efficient backup provisioning. In: 2018 14th International Conference on Network and Service Management (CNSM), 2018, pp. 108–115 (2018)
10. Dinh, N.-T., Kim, Y.: An efficient availability guaranteed deployment scheme for IoT service chains over fog-core cloud networks. Sensors **18**(11), 3970 (2018)
11. Zhu, H., Huang, C.: EdgePlace: availability-aware placement for chained mobile edge applications. Trans. Emerg. Telecommun. Technol. **29**(11), e3504 (2018)
12. Yala, L., Frangoudis, P.A., Ksentini, A.: Latency and availability driven VNF placement in a MEC–NFV environment. In: 2018 IEEE Global Communications Conference (GLOBECOM), 2018, pp. 1–7 (2018)
13. Zhang, Y., Zhao, Z., Shu, C., Min, G., Wang, Z.: Embedding virtual network functions with backup for reliable large-scale edge computing. In: 2018 5th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2018 4th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), 2018, pp. 190–195. IEEE (2018)
14. Huang, M., Liang, W., Shen, X., Ma, Y., Kan, H.: Reliability-aware virtualized network function services provisioning in mobile edge computing. IEEE Trans. Mob. Comput. **19**(11), 2699–2713 (2019)
15. Li, J., Liang, W., Huang, M., Jia, X.: Providing reliability-aware virtualized network function services for mobile edge computing. In: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), 2019, pp. 732–741. IEEE (2019)
16. Qu, L., Khabbaz, M., Assi, C.: Reliability-aware service chaining in carrier-grade softwarized networks. IEEE J. Sel. Areas Commun. **36**(3), 558–573 (2018)
17. Qu, L., Assi, C., Khabbaz, M.J., Ye, Y.: Reliability-aware service function chaining with function decomposition and multipath routing. IEEE Trans. Netw. Serv. Manag. **17**(2), 835–848 (2020)
18. Fan, J., Guan, C., Zhao, Y., Qiao, C.: Availability-aware mapping of service function chains. In: IEEE INFOCOM 2017—IEEE Conference on Computer Communications, 2017, pp. 1–9 (2017)
19. Zhang, J., Wang, Z., Peng, C., Zhang, L., Huang, T., Liu, Y.: RABA: resource-aware backup allocation for a chain of virtual network functions. In: IEEE INFOCOM 2019—IEEE Conference on Computer Communications, 2019, pp. 1918–1926 (2019)
20. Liu, Y., Lu, Y., Qiao, W., Chen, X.: Reliability-aware service chaining mapping in NFV-enabled networks. ETRI J. **41**(2), 207–223 (2019)
21. Bijwe, S., Machida, F., Ishida, S., Koizumi, S.: End-to-end reliability assurance of service chain embedding for network function virtualization. In: 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV–SDN), 2017, pp. 1–4 (2017)
22. Alahmad, Y., Agarwal, A.: VNF placement strategy for availability and reliability of network services in NFV. In: 2019 Sixth International Conference on Software Defined Systems (SDS), 2019, pp. 284–289 (2019)
23. Zhai, D., Meng, X., Yu, Z., Han, X.: Reliability-aware service function chain backup protection method. IEEE Access **9**, 14660–14676 (2021)
24. Fan, J., Jiang, M., Rottenstreich, O., Zhao, Y., Guan, T., Ramesh, R., Das, S., Qiao, C.: A framework for provisioning availability of NFV in data center networks. IEEE J. Sel. Areas Commun. **36**(10), 2246–2259 (2018)
25. Herker, S., An, X., Kiess, W., Beker, S., Kirstaedter, A.: Data-center architecture impacts on virtualized network functions service chain embedding with high availability requirements. In: 2015 IEEE GLOBECOM Workshops (GC Wkshps), 2015, pp. 1–7 (2015)

26. KaliyammalThiruvasagam, P., Kotagi, V.J., Murthy, S.R.: A reliability-aware, delay guaranteed, and resource efficient placement of service function chains in softwarized 5G networks. IEEE Trans. Cloud Comput. (2020). https://doi.org/10.1109/TCC.2020.3020269
27. Shang, X., Huang, Y., Liu, Z., Yang, Y.: Reducing the service function chain backup cost over the edge and cloud by a self-adapting scheme. In: IEEE INFOCOM 2020—IEEE Conference on Computer Communications, 2020, pp. 2096–2105 (2020)
28. IBM ILOG CPLEX Optimizer. Technical Report. https://www.ibm.com/analytics/cplex-optimizer

Thi-Thuy-Lien Nguyen received the Bachelor's and Master's Degrees in Computer Science from the Hanoi University of Technology, in 2011 and 2014, respectively. She is currently pursuing the Ph.D. Degree with the Department of Computer Science, VNU University of Engineering and Technology, Vietnam and holds a Faculty Position at the Department of Computer Science, Phenikaa University. Her research interests include the performance evaluation of computer networks and optimization problems in network functions virtualization.

Tuan-Minh Pham (Senior Member, IEEE) received the Ph.D. Degree in Computer Science from the Université Pierre et Marie Curie, France, in 2011. He was a Visiting Scientist with Pennsylvania State University and the Université Pierre et Marie Curie, in 2013 and 2017, respectively. He is currently working as a Faculty Member with the Faculty of Computer Science, Phenikaa University. His research interests include future internet architectures, modeling and analysis of networked systems, and network measurement for protocol evaluation. He was a recipient of the Best Paper Award from the IEEE International Conference on Social Computing, in 2013, and the Elsevier Computer Networks, in 2021. He has served as a Technical Committee Member and a Reviewer for the IEEE ICC, the IEEE CCNC, the IEEE LCN, IEEE Access, the Computer Communications (Elsevier), and the IEEE Transactions on Network and Service Management, among others.

Linh Manh Pham is a Lecturer at University of Engineering and Technology, Vietnam National University, Hanoi (VNU-UET). He was a Postdoctoral Researcher at Inria, France from 2016 to 2018. He earned an M.Sc. in Computer Science from the Texas A&M University System, USA and a Ph.D. in Cloud Computing from the Université Grenoble Alpes, France, in 2011 and 2016, respectively. His area of research is Cloud/Fog computing, analysis and design of computer networks as well as network performance evaluation. He has served as a Technical Committee Member and a Reviewer for the IEEE Cloud, the IEEE Transactions on Computers, the Euro-Par, the IEEE RIVF, along with others.