Reliability Augmentation of Requests with Service Function Chain Requirements in Mobile Edge-Cloud Networks

Weifa Liang
The Australian National University
Canberra, Australia
wliang@cs.anu.edu.au

Yu Ma
The Australian National University
Canberra, Australia
yu.ma@anu.edu.au

Wenzheng Xu
Sichuan University
Chengdu, P.R. China
wenzheng.xu@scu.edu.cn

Xiaohua Jia
City University of Hong Kong
Hong Kong, P. R. China
csjia@cityu.edu.hk

Sid Chi-Kin Chau
The Australian National University
Canberra, Australia
sid.chau@anu.edu.au

ABSTRACT

Provisioning reliable network services for mobile users in a mobile edge computing environment is the top priority for most network service providers, as unreliable or severely failed services will result in tremendous loss on their revenues and consumers. In this paper, we study a novel service reliability augmentation problem in a Mobile Edge-Cloud (MEC) network, where mobile users request various network services through issuing requests with service function chain (SFC) requirements and reliability expectations, and an admitted request may not meet its reliability expectation initially. To enhance its service reliability to reach its expectation, it is a common practice to make use of redundant backups, that is to place redundant VNF instances of each Virtual Network Function (VNF) in its SFC in case its primary VNF instance fails. In this paper, we aim to augment the reliability of each admitted request as much as possible with the ultimate objective to reach its reliability expectation, subject to computing capacity on each cloudlet in the network. To this end, we first formulate a novel service reliability augmentation problem. We then deal with the problem for the admitted request under the assumption that all the secondary VNF instances of each primary VNF instance in its SFC must be placed into the cloudlets no more than l hops from the cloudlet of the primary VNF instance, where $1 \leq l \leq n-1$ and n is the number of cloudlets in the network, for which we propose an integer linear program (ILP) solution, and develop a randomized algorithm with a provable approximation ratio while a moderate resource constraint violation. We also devise an efficient heuristic algorithm for the problem without any resource constraint violation. We finally evaluate the performance of the proposed algorithms through experimental simulations. Experimental results demonstrate that the proposed algorithms are promising, and their empirical results are superior to their analytical counterparts.

ICPP '20, August 17–20, 2020, Edmonton, AB, Canada
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-8816-0/20/08…$15.00
https://doi.org/10.1145/3404397.3404452

KEYWORDS

Reliability augmentation for service function chain provisioning; virtualized network function placement; approximation algorithms; Mobile Edge Computing; primary and secondary VNF instance placement.

ACM Reference Format:
Weifa Liang, Yu Ma, Wenzheng Xu, Xiaohua Jia, and Sid Chi-Kin Chau. 2020. Reliability Augmentation of Requests with Service Function Chain Requirements in Mobile Edge-Cloud Networks. In *49th International Conference on Parallel Processing - ICPP (ICPP '20), August 17–20, 2020, Edmonton, AB, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3404397.3404452

1 INTRODUCTION

Network Function Virtualization (NFV) and Mobile Edge Computing (MEC) have been envisioned as key enabling technologies to support delay-sensitive applications in smart cities, IoTs, and intelligent transportation. NFV decouples network functions from dedicated hardware - middleboxes, leading to significant cost reduction in network service provisioning. Network service providers provide mobile users with low-latency, highly reliable network services through the placement of VNFs to cloudlets in an MEC network to meet user service demands with service function chain (SFC) and reliability expectation requirements. Due to the chaining nature and distributed placement of VNF instances, the failure of any single VNF instance in a chain will heavily affect the normal operation of a service, and will result in serious data loss and resource waste.

In this paper, we consider reliability-aware network services provisioning in an MEC environment, where each mobile user issues a network service request with an SFC enforcement and a given reliability expectation. To improve user experience on the use of the virtualized services while meeting their reliability expectations ultimately, the deployment of redundant VNF instances is a common choice, by placing multiple redundant VNF instances for each network function in the SFC to different cloudlets. We distinguish between the single primary VNF instance and one or multiple secondary VNF instances for each network function [8]: the former is an active VNF instance while the latter are idle ones until the primary one fails. Consider limited resources in an MEC network, how to augment the reliability of an admitted request poses great challenges. For example, how many secondary VNF instances of each primary VNF instance in the service chain of a request need be instantiated? and to which cloudlets will these VNF instances be placed?

The novelties of the work in this paper lie in the formulation of a novel service reliability augmentation problem of each admitted request with a service function chain enforcement and a given reliability expectation in an MEC environment. We devise a non-trivial integer linear program solution and the very first randomized algorithm for the problem.

The main contributions of this paper are presented as follows.

- We formulate a novel service reliability augmentation problem for an admitted request with SFC and reliability expectation requirements in an MEC network, and show that the problem is NP-hard.
- We propose a non-trivial integer linear program (ILP) solution to the service reliability augmentation problem, under the assumption that all secondary VNF instances of each primary VNF instance must be placed into the cloudlets no more than l hops from the cloudlet of the primary VNF instance, where l is fixed with $1 \leq l \leq |V| - 1$. The value of l is used to control the latency of updating its secondary VNF states if there is any update on a primary VNF instance.
- We develop a randomized algorithm with a provable approximation ratio for the problem with high probability through linear relaxation on the ILP. However this is achieved at the expense of moderate violations of computing capacities on cloudlets.
- We devise an efficient heuristic algorithm for the service reliability augmentation problem without any resource violations, through a series of reductions of the problem to minimum-cost maximum matching problems in auxiliary graphs.
- We evaluate the performance of the proposed algorithms through experimental simulations, and the experimental results demonstrate that the proposed algorithms are promising and outperform their analytical counterparts.

The rest of the paper is organized as follows. Section 2 summarizes the related work of reliable service function provisioning. Section 3 introduces notions, notations, and the problem definitions. The NP-hardness of the problems is also shown in this section. Section 4 formulates an ILP solution for the service reliability augmentation problem. Section 5 develops a randomized approximation algorithm and Section 6 devises an efficient heuristic algorithm for the problem. Section 7 evaluates the proposed algorithms empirically, and Section 8 concludes the paper.

2 RELATED WORK

There are intensive efforts on the reliability (or availability) of VNF provisioning in data-center networks and MEC networks in the past several years, and a recent survey on this topic is given by Han *et al.* [8]. For example, Fan *et. al* [5, 6] studied the availability issue of service function chains. They proposed heuristic algorithms that map SFCs to servers in data center networks with the aim of minimizing the amounts of on-site and off-site backups required, in order to meet the given availability requirements. Fan *et al.* [7] considered the reliable-SFC instance service providing in a data center network, where there are sufficient computing resource for VNF instance placement, and all VNF instances (both primary and secondary VNF instances) of a function service chain is consolidated into a single server. Qu *et. al* [17] jointly considered the availability and delay constraints in the backup resource allocation problem of SFC with an objective to minimize the amount of bandwidth resource needed in a data center network, and they later extended their work by allowing the sharing of VNF instances in [18]. Ding *et al.* [3] formulated how to calculate VNF placement availability when at most one backup chain is allowed. They proposed a heuristic to find the backup SFCs with the minimum cost such that the accumulative availability for each request is met. Aidi *et al.* [1] proposed a framework to efficiently manage survivability of service function chains and the backup VNFs, with the aim to determine both the minimum number and optimal locations of backup VNFs to protect service function chains. They proposed heuristics for the problem.

There are also several works focusing on robust service provisioning in MEC, where each request has only a single service function rather than a service function chain requirement. For example, Huang *et al.* [11] studied the robust network function service provisioning in MEC environments, for which they developed approximation algorithms for primary and secondary VNF instance placements among MEC cloudlets. Under an ideal assumption that both network function and server failure probabilities are given, the backup VNF instances of any function should be placed into the same server, and all different functions have the same computing resource demands. He *et al.* [9] considered the assignment of backup VNF instances to different servers such that the maximum failure probability of the functions is minimized, and they provided two heuristic algorithms for the problem. Li *et al.* [12, 13] investigated the VNF instance placement of dynamic requests with a single VNF request with the aim to meet individual requests' reliability requirements. They devised an online algorithm with a constant competitive ratio for the problem when all VNF instances of the VNF of an admitted request are consolidated into a single cloudlet. However, they assumed that the primary and backup VNF instances can be placed to any cloudlets as long as there are sufficient computing resources in these cloudlets to accommodate the VNF instances. However, all of the mentioned studies focused on requests with a single VNF service, not a service function chain that consists of multiple different VNFs, and none of the studies has ever considered the service reliability augmentation issue on admitted requests through the placement of redundant VNF instances to different cloudlets. Lin *et al.* [14] recently studied the primary and backup VNF instance placement for a service chain in an MEC to meet the specified reliability requirement of each request, for which they proposed an randomized algorithm and a heuristic algorithm. However, they assumed that the primary and backup VNF instances can be placed to any cloudlets as long as there are sufficient computing resources in the cloudlets to accommodate the VNF instances.

Unlike the aforementioned studies that either conducted in datacenter networks or MEC networks, in this paper we study the provisioning of reliable virtualized network services through improving the service reliability. We focus on the service reliability augmentation problem of an admitted request through redundant VNF instance placement of different network functions, subject to the computing capacity on each cloudlet, under a more realistic assumption that the redundant VNF instances can only be placed into the cloudlets no more than l-hops from the cloudlets of their primary VNF instances in the SFC with $l = 1, \ldots, |V| - 1$.

3 SYSTEM MODEL

Consider that the MEC network is an undirected graph $G = (V, E)$, where V is the set of nodes and E is the set of links between nodes. Each node $v \in V$ is an Access Points (APs), which may or may not be co-located with a cloudlet. If it is co-located with a cloudlet, the computing capacity of the cloudlet is $C_v > 0$, otherwise, its computing capacity $C_v = 0$. Let $N_l(v)$ be the l-neighbor set of node v in G, where $N_l(v) = \{u \mid$ the distance between u and v is no greater than l hops in $G\}$ with $1 \leq l \leq |V| - 1$. Denote by $N_l^+(v) = N_l(v) \cup \{v\}$.

3.1 Request admission and its reliability utility function

Let SFC_j of request j consists of L_j different network functions $f_1, f_2, \ldots, f_{L_j}$ in order, where the reliability of any VNF instance of network function f_i is a given value r_i with $0 < r_i \leq 1$. Assuming that request j has been admitted, all the primary VNF instances of its SFC_j have been placed, and the reliability $\Pi_{l=1}^{L_j} r_l$ of the placed SFC_j may or may not be less than the reliability expectation ρ_j of request j. If the achieved reliability is strictly less than ρ_j, then we aim to augment its reliability as much as possible to reach the goal ρ_j, subject to the computing resource capacity on each cloudlet in G. Note that such a goal may never be reached due to the lack of demanded computing resource in the MEC network. In the following we show how to calculate the reliability of an admitted request j.

Let R_i be the reliability of function f_i in SFC_j by placing numerous its VNF instances (both primary and secondary VNF instances) to cloudlets. The calculation of R_i is as follows.

Assuming that there are n_i VNF instances of function $f_i \in \mathcal{F}$ instantiated in p cloudlets with $1 \leq p \leq n_i$, and let $r_{i,1}, r_{i,2}, \ldots, r_{i,n_i}$ be their reliabilities in these cloudlets, respectively. The accumulative reliability R_i of f_i then is

$$R_i = 1 - \Pi_{l=1}^{n_i}(1 - r_{i,l}). \qquad (1)$$

For the sake of convenience, in the rest of discussion, we assume that $r_{i,l} = r_i$ for all l with $1 \leq l \leq n_i$, i.e., the reliability of a VNF instance of f_i placed at different cloudlets is identical. This assumption has been widely adopted in literature [5, 6, 9, 11].

Through the placement of both primary and secondary VNF instances of each network function in SFC_j, the reliability u_j of request j is $u_j = \Pi_{i=1}^{L_j} R_i$.

To ensure that the reliability expectation ρ_j of request j can be achieved if there are sufficient resources in MEC, we have $\Pi_{i=1}^{L_j} R_i \geq \rho_j$, which can be equivalently written as follows.

$$-\sum_{i=1}^{L_j} \log R_i \leq -\log \rho_j. \qquad (2)$$

In other words, if ρ_j is not achievable due to lack of computing resource on each cloudlet in G, we aim to maximize the value of u_j, or minimize the value of $-\log u_j = -\sum_{i=1}^{L_j} \log R_i$.

3.2 Problem definition

Given an MEC network $G(V, E)$, each cloudlet $v \in V$ has a computing capacity C_v, and the set of network functions $\mathcal{F} = \{f_1, f_2, \ldots, f_{|\mathcal{F}|}\}$, each function $f_i \in \mathcal{F}$ needs $c(f_i)$ computing resource for its implementation in a VM with $1 \leq i \leq |\mathcal{F}|$. Let r_i be the reliability of f_i in any cloudlet $v \in V$ with $0 < r_i \leq 1$, assume that request j with a service function chain SFC_j has been admitted in G and its reliability expectation is a given value ρ_j. The reliability enhancement for request j is achieved through redundant VNF instance placements in cloudlets, *the service reliability augmentation problem* for an admitted request j thus is to maximize its reliability through deploying as many as secondary VNF instances of each VNF instance in SFC_j until its reliability expectation ρ_j is reached, or reaching its best possible reliability due to running out of computing resources of G.

Assuming that request j has been admitted, and all VNF instances of the functions in its SFC_j have been placed, we term these VNF instances as the *primary VNF instances* of the request. Notice that the primary VNF instance usually is in active status and all the *secondary VNF instances* of the primary VNF instances are in idle statuses. The primary VNF instance communicates with its secondary VNF instances at some pre-defined checking points to update its execution image/status to its secondary VNF instances, and we assume that such communication delay is negligible. To reduce the response delay of such updating, the secondary VNF instances usually are co-placed with its primary VNF instance either in the same cloudlet v or in no more than l-hop cloudlets in $N_l(v)$ from the cloudlet v of its primary VNF instance, where $l = 1, 2, \ldots, |V| - 1$.

3.3 NP hardness of the defined problem

Theorem 3.1. *The service reliability augmentation problem for an admitted request with a SFC and a reliability expectation in an MEC network $G = (V, E)$ is NP-hard.*

Proof. We reduce the minimum-cost generalized assignment problem (GAP) [16] to a special case of the service reliability augmentation problem where the reliability expectation of each request will not be considered. We term this special problem as Problem **P1** for simplicity.

We start with the definition of the minimum-cost GAP as follows. Given n items $a_1, a_2, \ldots, a_n$ and m bins, each item a_i has size $s(a_i)$ and each bin j has a capacity B_j. If item a_i is packed to bin j, it incurs a cost $c_{ij} > 0$ with $1 \leq i \leq n$ and $1 \leq j \leq m$, the problem is to pack as many as items to the m bins such that the total cost of the packed items is minimized, subject to the capacity on each bin. It is well known that the minimum-cost GAP is NP-hard [16].

We here reduce the minimum-cost GAP to Problem **P1** as follows. For the given n items, we assume that there is a request with a SFC that consists of n network functions $f_1, f_2, \ldots, f_n$, where network function f_i corresponds to item a_i with $1 \leq i \leq n$. We further assume the cloudlets in G are indexed by $1, 2, \ldots, m$ with $m = |V|$, and cloudlet j has a residual computing capacity B_j. Placing the secondary VNF instances of f_i to cloudlet j incurs a cost c_{ij}, while the total computing demand of the VNF instances of f_i is $s(a_i)$. We assume that G is a complete graph, then any secondary VNF instance of a primary VNF instance of f_i can be placed to any cloudlet u, i.e., any VNF instance of f_i can be placed to any cloudlet in G if the cloudlet has sufficient computing resource for it. The decision version of Problem **P1** is to determine whether all VNF instances of network functions in the SFC of the request can be placed to the m cloudlets while the total placement cost is minimized (or equivalently, the reliability achieved of the request is maximized), subject to the computing capacity on each cloudlet. It can be seen that if there is a solution to Problem **P1**, there is a solution to the minimum-cost GAP. It is known that the minimum-cost GAP is NP-hard, Problem **P1** thus is NP-hard. Since Problem **P1** is a special case of the service reliability augmentation problem, the latter is NP-hard, too. □ □

4 INTEGER LINEAR PROGRAMMING

In this section, we first provide an initial admission framework of a single request j with SFC_j and reliability expectation ρ_j. We then formulate the service reliability augmentation problem as an integer linear program (ILP).

4.1 An initial admission framework of a request with an SFC requirement

We provide an initial admission framework of a single request j with SFC_j and reliability expectation ρ_j, by instantiating its primary VNF instances of network functions in SFC_j to the cloudlets in G such that the reliability of the admitted request is maximized. Such an admission provides a basic reliability for the request. At this stage, we do not consider instantiating any of secondary VNF instances yet.

We adopt the similar technique in [15] for the admission of request j. That is, we construct an auxiliary directed acyclic graph (DAG) $G_j = (N_j \cup \{s_j, t_j\}, A_j; \omega)$, where N_j is the set of cloudlets to host the primary VNF instances of network functions in SFC_j, s_j and t_j are the source and destination cloudlets of data traffic of request j respectively. A_j is the set of edges in G_j from one cloudlet to another cloudlet if there is a path between the two cloudlets in G. Function $\omega : E \mapsto [0, 1]$ is a reliability weight function on the edges of G_j. A shortest path in G_j then corresponds to a placement of the primary VNF instances of SFC_j for request j with the maximum reliability, the details can be referred to the paper [15].

4.2 Reliability augmentation of an admitted request

We consider the service reliability augmentation problem for an admitted request j, where all the secondary VNF instances of each primary VNF instance in SFC_j must only be placed no more than l-hops cloudlet of the cloudlet of the primary VNF instance. For the sake of convenience, in the following we only focus on the case where $l = 1$, the rest (with $l > 1$) is almost identical with the case $l = 1$, and omitted.

Assume that the primary VNF instance of f_i in SFC_j is placed in cloudlet $v \in V$. Let $N_1(v) = \{u_1, u_2, \ldots, u_{d_v}\}$, where d_v is the number of one-hop neighbor cloudlets of cloudlet v in $G(V, E)$ with residual computing capacities $C'_{u_1}, C'_{u_2}, \ldots, C'_{u_{d_v}}$, respectively. Assuming that cloudlet v is cloudlet u_0, i.e., $C'_v = C'_{u_0}$. Let $k_{i,l} = \lfloor \frac{C'_{u_l}}{c(f_i)} \rfloor$ with $0 \leq l \leq d_v$. For network function f_i in SFC_j, there are at most $d_v + 1$ bins with bin u_l having the residual computing capacity C'_{u_l} and $0 \leq l \leq d_v$, there are at most $K_i = \sum_{l=0}^{d_v} k_{i,l}$ items of type f_i with each representing one potential secondary VNF instances of f_i.

For each item k_i of type f_i with $1 \leq k_i \leq K_i$, assume that its primary VNF instance is placed in cloudlet v, then the cost of item k_i placed at any cloudlet $u \in N_l^+(v)$ is $c(f_i, k_i, u) = -\log(R(f_i, k_i) - R(f_i, k_i - 1))$; otherwise, the cost $c(f_i, k_i, u)$ of item k_i placed to cloudlet u is $c(f_i, k_i, u) = M$ for any $u \notin N_l^+(v)$, where M is a sufficiently large positive number, e.g., $M = 100 * \max\{c(f_i, k_i, u) \mid u \in N_l(v) \cup \{v\},\ 0 \leq k_i \leq K_i,\ 1 \leq i \leq L_j\}$, the amount of the computing resource consumed by item k_i is $c(f_i)$. Since there are L_j different primary VNF instances for SFC_j, there are L_j different types of items.

We then reduce the service reliability augmentation problem for an admitted request j with SFC_j and reliability expectation ρ_j to *a budgeted minimum cost generalized assignment problem* (BMCGAP) that is defined as follows.

Given m bins indexed by $1, 2, \ldots, m$ with the bin capacity B_j of bin j, and a set $\mathcal{I}$ of n items, each item $I_i \in \mathcal{I}$ has a positive cost c_{ij} with size s_{ij} if item I_i is packed to bin j, assume that the total cost budget C is given, the problem is to pack as many as items in $\mathcal{I}$ to the m bins such that the total cost of packed items is minimized but upper bounded by C, subject to the capacity on each bin.

4.3 Overview of the proposed algorithm

The reduction proceeds as follows. There are $|V|$ bins with each $v \in V$ having the residual computing capacity of C'_v if $C'_v \neq 0$. Let K_i $(= \sum_{u \in N_l^+(v)} \lfloor \frac{C'_u}{c(f_i)} \rfloor)$ be the maximum number of secondary VNF instances of f_i that can be placed in one or multiple cloudlets u in $N_l^+(v)$ if the cloudlets have sufficient computing resource to accommodate the VNF instances, assuming that the primary VNF instance of f_i is in cloudlet v. Denote by $N_{f,v}$ the set of different types of primary VNF instances of SFC_j placed in cloudlet v.

For each network function f_i in SFC_j, there are K_i items of type f_i with the same computing resource demand $c(f_i)$, they can be placed to at most $d_v + 1$ bins under the assumption of the secondary VNF instance placement, i.e., the bins in $N_l^+(v)$ if $f_i \in N_{f,v}$. However, different items of type f_i will incur different costs, i.e., item k_i of type f_i will incur a cost $c(f_i, k_i, u)$ defined in Eq. (3) if it is placed to bin $u \in N_l^+(v)$; otherwise, it will incur a cost $c(f_i, k_i, u) = M$, where M is defined as a large positive value with $0 \leq k_i \leq K_i$ and $1 \leq i \leq L_j$. There are L_j different types of items.

$$c(f_i, k, u) = -\log(R(f_i, k) - R(f_i, k - 1)), \quad (3)$$
$$1 \leq k \leq K_i, u \in N_l^+(v), \text{ and } f_i \in N_{f,v},$$
$$c(f_i, 0, v) = -\log R(f_i, 0), \quad \text{if } f_i \in N_{f,v} \text{ and } v \in V, \quad (4)$$

where $R(f_i, 0) = r_i$, and $R(f_i, k) = 1 - (1 - r_i)^{k+1}$.

The BMCGAP thus is to pack as many as items in $\mathcal{I}$ into the $|V|$ bins to minimize the total cost, subject to the cost budget C $(= -\log \rho_j)$ and the residual computing capacity C'_v on each cloudlet $v \in V$.

4.4 Integer linear program formulation

In the following, we propose an ILP solution to the service reliability augmentation problem for an admitted request j. By Ineq. (2), the optimization objective is to

$$\text{minimize} \quad \sum_{i=1}^{L_j} -\log R_i \tag{5}$$

$$\text{subject to:} \tag{6}$$

$$-\log R_i = \sum_{k_i=0}^{K_i} c(f_i, k_i, u) \cdot x_{i,k_i,u}, \tag{7}$$

$$f_i \in N_{f,v},\ u \in N_l^+(v), \text{ and } 1 \le i \le L_j$$

$$\sum_{u \in N_l^+(v)} x_{i,k_i,u} \le 1,\ f_i \in N_{f,v} \text{ and } 1 \le i \le L_j \tag{8}$$

$$\sum_{i=1}^{L_j} \sum_{k_i=0}^{K_i} c(f_i) \cdot x_{i,k_i,u} \le C'_u,\ u \in V \text{ and } f_i \in N_{f,v} \tag{9}$$

$$x_{i,k_i,u} \in \{0, 1\}, \tag{10}$$

$$x_{i,k_i,u} = 0, \quad \text{if } C'_u < c(f_i), \tag{11}$$

$$x_{i,k_i,u} = 0, \quad \text{if } u \in V \setminus N_l^+(v) \;\&\; f_i \in N_{f,v}, \tag{12}$$

$$x_{i,k_i,u} = 0, \quad \text{if } c(f_i, k_i, u) = M, \tag{13}$$

where R_i is the achieved reliability of network function $f_i \in SFC_j$ through placing multiple VNF instances to different cloudlets, which is defined in Eq. (1), and K_i is the number of secondary VNF of f_i with $K_i = \sum_{u \in N_l^+(v)} \lfloor \frac{C'_u}{c(f_i)} \rfloor$, $1 \le i \le L_j$ and $f_i \in N_{f,v}$. Note that when R_i becomes larger through placing more its VNF instances into the network, the value of $-\log R_i$ (> 0) becomes smaller and $0 < R_i \le 1$. Variable $x_{i,k_i,u}$ is a binary variable, if it is 1, then, the k_ith secondary VNF instance of f_i is placed to cloudlet $u \in V$.

Constraint (8) ensures that each item can be placed to no more than one cloudlet. Constraint (9) ensures that different secondary VNF instances at each cloudlet u is no more than its capacity. Constraint (11) ensures that none of any secondary VNF instance is placed to a cloudlet without its demanded computing resource. Constraint (12) ensures that any secondary VNF instance of a primary VNF instance placed in cloudlet v will not be placed to a cloudlet with more than l-hops from the cloudlet of its primary VNF instance. Constraint (13) is equivalent to Constraint (11), which implies that the VNF instance corresponding item k_i of type f_i cannot be placed into cloudlet u.

4.5 Algorithm analysis

In the rest, we first show the property of the cost function $c(\cdot)$ defined in (3) by the following lemma. We then analyze the property of the exact solution of the ILP.

Lemma 4.1. *For the defined cost function in Eq.* (3), *we have*

$$(1)\ c(f_i, k, u) > 0,\ \forall k \ge 0 \text{ and } f_i \in N_{f,u}, \tag{14}$$

$$(2)\ c(f_i, k', *) > c(f_i, k, *), \text{ if } k' > k \ge 1,\ f_i \in N_{f,v}, \text{ and } * \text{ is any cloudlet in } N_l^+(v). \tag{15}$$

Proof. When $k = 0$, $c(f_i, 0, u) = r_i > 0$.

When $k \ge 1$, $R(f_i, k) = 1-(1-r_i)^{k+1}$ and $R(f_i, k-1) = 1-(1-r_i)^k$, we then have $R(f_i, k) > R(f_i, k-1)$ due to $0 < r_i < 1$ and $c(f_i, k, u) = -\log(R(f_i, k) - R(f_i, k-1)) > 0$.

We now show that $c(f_i, k', *) > c(f_i, k'-1, *)$ as follows.

$$\begin{aligned} & c(f_i, k', *) - c(f_i, k'-1, *) \\ =\ & -\log(R(f_i, k') - R(f_i, k'-1)) \\ & -(-\log(R(f_i, k'-1) - R(f_i, k'-2))) \\ =\ & \log(R(f_i, k'-1) - R(f_i, k'-2)) \\ & -\log(R(f_i, k') - R(f_i, k'-1)) \\ =\ & \log \frac{1}{(1-r_i)} \\ & > 0, \quad \text{since } \frac{1}{1-r_i} > 1. \end{aligned} \tag{16}$$

By Ineq. (16), we have

$$c(f_i, k', *) > c(f_i, k'-1, *) > \ldots > c(f_i, k, *), \text{ if } k' > k. \tag{17}$$

The lemma then follows. □ □

Lemma 4.2. *Given an exact solution to the ILP, we claim that if* $x_{i,k_i,*} = 1$ *with* k_i *is the largest value in the solution that is no greater than* K_i, *then* $x_{i,k'_i,*} = 1$ *for any* $k'_i \le k_i$, *where* $*$ *represents any cloudlet* $u \in N_l^+(v)$ *and* $f_i \in N_{f,v}$.

Proof. We show the claim by contradiction. Assume that there exists $x_{i,k_i,u} = 1$ while $x_{i,k'_i,u'} = 0$ with $k'_i < k_i$ in the solution with $u' \in N_l^+(v)$. Following the cost definition and Lemma 4.1, $c(f_i, k_i, u) > c(f_i, k'_i, u')$ but both items k'_i and k_i have the same size $c(f_i)$. Another better solution with a smaller cost can be obtained, by replacing item k_i with item k'_i. This contradicts the fact that the solution obtained by the ILP is the optimal one with the minimum cost. The lemma then follows. □ □

5 RANDOMIZED ALGORITHM

In this section, we devise a randomized algorithm for the service reliability augmentation problem based on the ILP formulation.

5.1 Randomized algorithm description

Following the random rounding technique [19], we first relax the ILP to a Linear Program (LP), and an optimal solution for the LP can be obtained in polynomial time. We then round the fractional solution of the LP with probability to a 0/1 integer solution. We finally show that the 0/1 integer solution is very likely to be a feasible solution of the ILP with high probability.

The detailed randomized algorithm, `Algorithm` 1, for the service reliability augmentation problem of an admitted request j is given as follows.

5.2 Algorithm analysis

The rest is to analyze the approximation ratio of `Algorithm` 1 and the computing resource violation on each cloudlet. We start with the following lemma.

Lemma 5.1. *(Chernoff bounds) Given* n *independent variables* $x_1, x_2, \ldots, x_n$, *where* $x_i \in [0, 1]$. *Let* $\mu = \mathbb{E}[\sum_{i=1}^n x_i]$. *Then,*

```
Algorithm 1: A randomized algorithm for the service reliability augmentation problem
input : An MEC network $G = (V, E)$, and request $j$ with $SFC_j$ and reliability expectation $\rho_j$, assuming that
        the primary VNF instances of $SFC_j$ have been placed into the cloudlets in $G$ through invoking the
        initial admission framework for request $j$.
output: Find a solution for the problem, where all the secondary VNF instances of each primary VNF
        instance will be placed to the cloudlets no more than $l$-hops from the cloudlets of their primary
        VNF instances to maximize the reliability of request $j$ until either reaching its reliability
        expectation $\rho_j$ or as large as possible.
begin
    if the reliability of the admission of request j, $\Pi_{l=1}^{L_j} r_l \geq \rho_j$ then
        EXIT;
    Solve the relaxed version LP of ILP (5) in polynomial time. Let $\tilde{OPT}$ be the optimal solution of the LP and
      $\tilde{x}_{i,k_i,u}$ the value of each variable $x_{i,k_i,u}$, where $\tilde{x}_{i,k_i,u} \in [0,1]$;
    An integer solution $\hat{x}_{i,k_i,u}$ can be obtained by the randomized rounding approach in [19]. That is, $\hat{x}_{i,k_i,u}$
      is set to 1 with probability of $\tilde{x}_{i,k_i,u}$; otherwise, $\hat{x}_{i,k_i,u}$ is set to 0; The choice is performed in an exclusive
      manner, with Constraint (8): for each $u$, $\forall u \in N_l^+(v)$, exactly one of the variables $\hat{x}_{i,k_i,u}$ is set to one 1 and
      the rest is set to 0. This random choice is made independently for all such $u$s;
    A candidate integer solution $\hat{S}$ can be derived based on $\hat{x}_{i,k_i,u}$, which will be a feasible solution to the ILP with
      high probability.
```

(i) *Upper Tail:* $Pr[\sum_{i=1}^{n} x_i \geq (1+\beta)\mu] \leq e^{\frac{-\beta^2\mu}{2+\beta}}$ *for all* $\beta > 0$,

(ii) *Lower Tail:* $Pr[\sum_{i=1}^{n} x_i \leq (1-\beta)\mu] \leq e^{\frac{-\beta^2\mu}{2}}$ *for all* $0 < \beta < 1$.

We then have the following theorem.

Theorem 5.2. *Given an MEC network $G(V,E)$ and an admitted request j with SFC_j and reliability expectation ρ_j, there is a randomized algorithm,* Algorithm *1, with high probability of* $\min\{1-\frac{1}{N}, 1-\frac{1}{|V|^2}\}$*, for the service reliability augmentation problem. The expected approximation ratio of the algorithm is* $(1/P^*)^{1-\frac{2}{\Lambda}}$*, and the computing resource violation ratio at any cloudlet is no more than twice its capacity provided that* $P^* \geq \frac{1}{N^{3\Lambda/\log e}}$ *and* $min_{v \in V}\{C_v\} \geq 6\Lambda \ln V$*, where* $N = \sum_{i=1}^{L_j} K_i \leq \lceil \frac{L_j \cdot C_{max} \cdot d_{max}}{c_{min}} \rceil$*, K_i is the maximum potential number of secondary VNF instances for network function $f_i \in SFC_j$, Λ is a constant strictly greater than 2, and P^* is the optimal reliability of request j in G.*

Proof. We first analyze the approximation ratio of the randomized algorithm. We then show that the computing resource violation at each cloudlet is no more than twice its computing capacity. Denote by Λ a given value, which is defined as follows.

$$\Lambda = \max\{\max\{c(f_i,k_i,*) \mid I_{k_i} \in \mathcal{I}\}, \max\{C'_u \mid u \in V\}, \max\{c(f_i) \mid f_i \in SFC_j\}, -\log \rho_j\}. \quad (18)$$

Let $\tilde{OPT}$ be the optimal solution of the linear program (LP). Clearly, the value of $\tilde{OPT}$ is a lower bound on the value of the optimal solution OPT of the ILP. Recall that $\tilde{x}_{i,k_i,u}$ are the values of variables for the solution of the LP, which are within $[0,1]$.

Denote by $y_{i,k_i,u}$ a random variable derived from the random variable $x_{i,k_i,u}$, and the value of $y_{i,k_i,u}$ is $\frac{c(f_i,k_i,u)}{\Lambda}$ with probability $\tilde{x}_{i,k_i,u}$. Thus, the value range of $y_{i,k_i,u}$ is within $[0,1]$ as $\frac{c(f_i,k_i,u)\cdot \tilde{x}_{i,k_i,u}}{\Lambda} \leq \frac{c(f_i,k_i,u)\cdot \tilde{x}_{i,k_i,u}}{\max_{I_{k_i}\in\mathcal{I}}\{c(f_i,k_i,*)\}} \leq 1$.

We treat the N $(= \sum_{i=1}^{L_j} K_i)$ random variables $y_{i,k_i,u}$ as independent random variables with value ranges in $[0,1]$. Then,

$$\mathbb{E}[\sum_{I_{k_i}\in\mathcal{I}} y_{i,k_i,u}] = \sum_{I_{k_i}\in\mathcal{I}} \frac{c(f_i,k_i,u)\cdot \tilde{x}_{i,k_i,u}}{\Lambda} = \frac{\tilde{OPT}}{\Lambda}. \quad (19)$$

Let $\mu = \mathbb{E}[\sum_{I_{k_i}\in\mathcal{I}} c(f_i,k_i,u)\cdot \tilde{x}_{i,k_i,u}] = \tilde{OPT}$. Following the Chernoff bound in Lemma 5.1 (i), we have

$$\begin{aligned}
&\mathbf{Pr}[\sum_{I_{k_i}\in\mathcal{I}, f_i\in N_{f,v}, u\in N_l^+(v)} c(f_i,k_i,u)x_{i,k_i,u} \geq (1+\beta)OPT]\\
\leq&\mathbf{Pr}[\sum_{I_{k_i}\in\mathcal{I}, f_i\in N_{f,v}, u\in N_l^+(v)} c(f_i,k_i,u)x_{i,k_i,u} \geq (1+\beta)\tilde{OPT}]\\
&\qquad\qquad \text{since } \tilde{OPT} \leq OPT\\
=&\mathbf{Pr}[\sum_{I_{k_i}\in\mathcal{I}, f_i\in N_{f,v}, u\in N_l^+(v)} y_{i,k_i,u} \geq (1+\beta)\frac{\tilde{OPT}}{\Lambda}]\\
\leq& exp(\frac{-\beta^2\cdot\mu}{2+\beta}) \quad \text{for all } \beta > 0. \qquad (20)
\end{aligned}$$

We then assume that

$$exp(\frac{-\beta^2\cdot\frac{\mu}{\Lambda}}{2+\beta}) \leq \frac{1}{N}, \quad (21)$$

$$\beta > 0. \quad (22)$$

We set $0 < \beta \leq 1$, Ineq. (21) is transformed as follows.

$$exp(\frac{-\beta^2\cdot\mu}{3\Lambda}) \leq \frac{1}{N}, \quad (23)$$

when $N = |\mathcal{I}|$ is sufficiently large, the solution of Ineq. (23) is

$$\beta \geq \sqrt{\frac{3\Lambda \ln N}{\mu}} = \sqrt{\frac{3\Lambda \ln N}{\tilde{OPT}}} \geq \sqrt{\frac{3\Lambda \ln N}{OPT}}. \quad (24)$$

Since $\beta \leq 1$, we must have $OPT \geq 3\Lambda \ln N$. Thus, the optimal reliability P^* of request j is at least $P^* \geq 2^{-OPT} = (\frac{1}{2^{\ln N}})^{3\Lambda} = \frac{1}{N^{3\Lambda/\log e}}$. The approximation ratio of the randomized algorithm, Algorithm 1, then is no more than $1+\beta = 2$ with high probability $1-\frac{1}{N} = \frac{1}{c'|V|}$, in terms of the optimization objective (5), where $N \leq \lceil L_j \cdot \frac{C_{max}\cdot d_{max}}{c_{min}} \rceil \leq \lceil L_j \cdot \frac{C_{max}}{c_{min}} \cdot |V| \rceil \leq c' \cdot |V|$ if L_j, C_{max}, c_{min} and d_{max} are constants.

From the approximate solution obtained for the optimization objective (5), we now derive an approximate solution to the service reliability augmentation problem as follows.

Let A be the value of the solution delivered by the randomized algorithm, then $A \leq \frac{2 \cdot OPT}{\Lambda}$. Since the original problem is to maximize the reliability of request j, we have $2^{-OPT} \geq P^*$, where P^* is the optimal reliability of the problem.

We then have

$$\frac{2^{-A}}{P^*} \geq \frac{P^{*\frac{2}{\Lambda}}}{P^*} = P^{*(\frac{2}{\Lambda}-1)} = \frac{1}{P^{*(1-\frac{2}{\Lambda})}}. \tag{25}$$

We finally analyze the computing resource violation on each cloudlet $u \in V$ in the solution delivered by the randomized algorithm. The analysis technique adopted is similar to the one for the approximation ratio analysis of the algorithm.

Let $z_{i,k_i,u}$ be a random variable derived from the random variable $x_{i,k_i,u}$ for each item $I_{k_i} \in \mathcal{I}$ and the value of $z_{i,k_i,u}$ be $\frac{c(f_i)}{\Lambda}$ with probability of $\tilde{x}_{i,k_i,u}$ if $f_i \in N_{f,v}$ and $u \in N_l^+(v)$. It can be seen that there are N random variables $z_{i,k_i,u}$ for all $I_{k_i} \in \mathcal{I}$, which are assumed to be independent random variables with the value ranges in $[0, 1]$.

$\mathbb{E}[z_{i,k_i,u}] = \frac{c(f_i) \cdot \tilde{x}_{i,k_i,u}}{\Lambda} \leq \frac{c(f_i)}{\max_{I_{k_{i'}} \in \mathcal{I}} \{c(f_{i'})\}} \leq 1.$

Let μ_1 be the computing resource consumption of that all corresponding items of the variables are packed to cloudlet $u \in N_l^+(v)$ among the N random variables $z_{i,k_i,u}$ $\forall I_{k_i} \in \mathcal{I}$ if $f_i \in N_{f,v}$ and $u \in N_l^+(v)$. Then,

$$\begin{aligned} \mu_1 &= \mathbb{E}[\sum_{I_{k_i} \in \mathcal{I}, f_i \in N_{f,v}, u \in N_l^+(v)} z_{i,k_i,u}] \\ &= \sum_{I_{k_i} \in \mathcal{I}, f_i \in N_{f,v}, u \in N_l^+(v)} \frac{c(f_i) \cdot \tilde{x}_{i,k_i,u}}{\Lambda} = \frac{\tilde{C}'_u}{\Lambda}, \end{aligned} \tag{26}$$

where $\tilde{C}'_u$ is the computing resource consumed at cloudlet u in the solution of the LP, and $\tilde{C}'_u \leq C_u$.

Since there are $|V|$ cloudlets, the probability of the computing capacity violation of any of the cloudlets is

$$\begin{aligned} &\mathbf{Pr}[\bigvee_{v \in V} \sum_{I_{k_i} \in \mathcal{I}, f_i \in N_{f,v}, u \in N_l(v) \cup \{v\}} c(f_i) x_{i,k_i,u} \geq (1+\beta_1) C'_u] \\ &\leq \mathbf{Pr}[\bigvee_{v \in V} \sum_{I_{k_i} \in \mathcal{I}, f_i \in N_{f,v}, u \in N_l^+(v)} c(f_i) x_{i,k_i,u} \geq (1+\beta_1) \tilde{C}'_u] \\ &\qquad \text{since } \tilde{C}'_u \leq C'_u \end{aligned} \tag{27}$$

$$\begin{aligned} &= \sum_{v \in V} \mathbf{Pr}[\sum_{I_{k_i} \in \mathcal{I}, f_i \in N_{f,v}, u \in N_l^+(v)} z_{i,k_i,u} \geq (1+\beta_1) \frac{\tilde{C}'_u}{\Lambda}] \\ &\leq |V| \cdot exp(-\frac{\beta_1^2 \cdot \mu_1}{2+\beta_1}), \quad \text{by the Chernoff bound.} \end{aligned} \tag{28}$$

We set $\beta_1 \leq 1$, and let

$$exp(-\frac{\beta_1^2 \cdot \mu_1}{2+\beta_1}) \leq \frac{1}{|V|^2}. \tag{29}$$

Then, since $\tilde{C}'_u \leq C'_u$, we have

$$\beta_1 \geq \sqrt{\frac{6 \ln |V|}{\mu_1}} = \sqrt{\frac{6\Lambda \ln |V|}{\tilde{C}'_u}} \geq \sqrt{\frac{6\Lambda \ln |V|}{C'_u}}. \tag{30}$$

As $\beta_1 \leq 1$, we must have $C'_u \geq 6\Lambda \ln |V|$. To ensure that $C'_u \geq 6\Lambda \ln |V|$ for any cloudlet $u \in V$, we have $\min\{C'_u \mid u \in V\} \geq 6\Lambda \ln |V|$.

We then have

$$\begin{aligned} &\mathbf{Pr}[\bigvee_{v \in V} \sum_{I_{k_i} \in \mathcal{I}, f_i \in N_{f,v}, u \in N_l(v) \cup \{v\}} c(f_i) x_{i,k_i,u} \geq (1+\beta_1) C'_u] \\ &\leq \mathbf{Pr}[\bigvee_{v \in V} \sum_{I_{k_i} \in \mathcal{I}, f_i \in N_{f,v}, u \in N_l^+(v)} c(f_i) x_{i,k_i,u} \geq (1+\beta_1) \tilde{C}'_u] \\ &\leq |V| \cdot exp(-\frac{\beta_1^2 \cdot \mu_1}{2+\beta_1}) \quad \text{by (29)} \end{aligned} \tag{31}$$

$$\leq |V| \cdot \frac{1}{|V|^2} = \frac{1}{|V|}. \tag{32}$$

The theorem then follows. □ □

6 HEURISTIC ALGORITHM

As the service reliability augmentation problem is NP-hard, the proposed ILP solution for it may only be applicable when the problem size is small or moderate. While the proposed randomized algorithm has a provable approximation ratio with high probability, it is at the expense of moderate violations on computing resource capacities. In this section, we propose an efficient heuristic algorithm for the problem that delivers a feasible solution without any computing capacity violations.

6.1 Overview of the algorithm

The basic idea is to augment the reliability of request j through constructing a series of bipartite graphs $G_0, G_1, \ldots, G_l$. For each bipartite graph G_l, find a minimum-cost maximum matching M_l from G_l, which corresponds to a subset of secondary VNF instance placement to their matched cloudlets without violating the computing capacity of any cloudlet. This procedure continues until either the total cost reaches the cost budget C, or no more computing resource is available for further secondary VNF instance placement.

6.2 Algorithm

Following the problem optimization objective, we aim to increase the reliability of request j by placing as many secondary VNF instances as possible to the cloudlets while minimizing their placements, subject to the cost budget C and computing resource capacity on each cloudlet.

To this end, we construct a series of auxiliary bipartite graphs. We start with graph $G_0 = (V, \mathcal{I}, E_0; c)$ as follows. Each node $v \in V$ has a residual computing capacity C'_v, and $\mathcal{I}$ is the set of all possible secondary VNF instances of VNFs in SFC_j, i.e., $\mathcal{I} = \cup_{i=1}^{L_j} \cup_{k_i=0}^{K_i} \{I_{k_i}\}$, there is an edge $(u, I_{k_i}) \in E_0$ in G_0 between nodes $u \in V$ and $I_{k_i} \in \mathcal{I}$ with cost $c(f_i, k_i, u)$ if $f_i \in N_{f,v}$, $u \in N_l^+(v)$, and $C'_u \geq c(f_i)$. The detailed algorithm is presented in `Algorithm` 2.

Algorithm 2: Heuristic algorithm for the service reliability augmentation problem

```
input  : An MEC network $G(V,E)$ with residual computing
         capacity $C'_v$ and an admitted request $j$ with the
         primary VNF instances of its $SFC_j$ placed and
         reliability expectation $\rho_j$.
output : Augment the reliability of request $j$ by placing all
         the secondary VNF instances of each primary VNF
         instance to the cloudlets no more than $l$-hops from
         the primary VNF instance, subject to the residual
         computing capacity on each cloudlet $v \in V$ and the
         total placement budget $C = -\log \rho_j$.
begin
    if $\Pi_{l=1}^{L_j} r_i \geq \rho_j$ then
        the admission of request $j$ meets its reliability
          expectation $\rho_j$;
        EXIT;
    Construct the initial bipartite graph $G_0(V, \mathcal{I}, E_0; c)$;
    $S \leftarrow \emptyset$; /* the solution */
    $l \leftarrow 1$; $G_1 \leftarrow G_0$; $E_1 \leftarrow E_0$;
    while ($c(S) < C$ and $E_l \neq \emptyset$) do
        Find a minimum-cost maximum matching $M_l$ in $G_l$,
          by the Hungarian algorithm;
        $S \leftarrow S \cup M_l$;
        $C'_v \leftarrow C'_v - c(f_i)$ if $\exists (v, I_{k_i}) \in M_l$ for each $v \in V$;
        $l \leftarrow l + 1$; $\mathcal{I} \leftarrow \mathcal{I} \setminus \{I_{k_i} \mid (v, I_{k_i}) \in M_l\}$;
        Construct the next bipartite graph
          $G_l = (V', \mathcal{I}, E_l; c)$, where
          $V' = \{v \mid v \in V$ and $C'_v \neq 0\}$; $E_l$ is the set of edges
          between the nodes in $V'$ and $\mathcal{I}$, and an edge
          $(v, I_{k_i}) \in E_l$ if $f_i \in N_{f,v}$ and $C'_v \geq c(f_i)$;
        $c(S) \leftarrow \sum_{(v, I_{k_i}) \in S} c(f_i, k_i, v)$; /* the total cost of the
          solution */
    return Solution $S$.
```

6.3 Algorithm analysis

In the following, we first show that the solution delivered by Algorithm 2 is feasible. We then analyze its time complexity.

Lemma 6.1. *For any function f_i in SFC_j of request j, assume that its primary VNF instance is placed at cloudlet v, if there are K'_i items of this type function that have been packed into cloudlets in $N_l^+(v)$ with $0 \leq K'_i \leq K_i$, by* Algorithm *2, then, these packed K'_i items must be the top-K'_i smallest items in terms of the defined cost.*

Proof. Assume that there is an item k_i for f_i which is placed in a bin $u \in N_l^+(v)$ that is not one of the first K'_i smallest items of this type. Let k'_i be one of the top-K'_i smallest items, i.e., $k'_i \leq K'_i$ while $k_i > K'_i$. We replace item k_i by item k'_i into bin u of item k_i, there does not incur any change in terms of the amounts of computing resource consumption for either of them. However, the amount of cost reduced by this replacement is $c(f_i, k_i, v) - c(f_i, k'_i, v) > 0$ by Lemma 4.1, as $k_i > k'_i$. The lemma then follows. □ □

Theorem 6.2. *Given an MEC network $G(V,E)$ and an admitted request j with SFC_j and reliability expectation ρ_j, each cloudlet $v \in V$ has residual computing capacity C'_v. There is an efficient algorithm,* Algorithm *2, for the service reliability augmentation problem for the admitted request j, under the assumption that all the secondary VNF instances of each primary VNF instance in SFC_j must be placed into the cloudlets no more than l-hops from the cloudlet of the primary VNF instance, where l is a fixed integer with $1 \leq l \leq |V| - 1$. The the time complexity of* Algorithm *2 is $O((N^3 + |V|^3) \cdot \log_{\frac{d_{min}}{d_{min^{+1}}}} N)$, where $N = \lceil \sum_{i=1}^{L_j} K_i \rceil \leq \lceil \frac{L_j \cdot C_{max} \cdot d_{max}}{c_{min}} \rceil$, $d_{min} = \min\{d_v \mid v \in V\}$, $d_{max} = \max\{d_v \mid v \in V\}$, $C_{max} = \max_{v \in V}\{C_v\}$, $c_{min} = \min\{c(f_i) \mid f_i \in SFC_j\}$, and $L_j = |SFC_j|$.*

Proof. We first show that the solution is feasible, That is, there is sufficient computing resource for each packed item (a secondary VNF instance) of type f_i to be instantiated, and none of the residual computing capacity on any cloudlet will be violated.

Following Algorithm 2, an item of type f_i is packed to a bin if the bin has residual computing capacity no less than its computing resource demand $c(f_i)$, i.e., the second VNF instance of f_i can be instantiated in the cloudlet. Furthermore, we claim that no secondary VNF instance of f_i will be placed to a cloudlet $v' \notin V \setminus N_l^+(v)$. Otherwise, even if there is such a placement, the cost by the placement is M, which is a very large number, in spite of it does consume the amount $c(f_i)$ of computing resource of cloudlet v'. We can remove this placement as it can reduce the total cost and save the amount of $c(f_i)$ computing resource in cloudlet v'. It is also noted that the total computing resource consumption of all packed items in any bin is no more than its capacity, which is implemented by the maximum matching M_l with $l \geq 1$, following the proposed algorithm, i.e., if there is a matched edge in G_l, then the corresponding VNF instance can be placed to that cloudlet following the edge construction. Following Lemma 4.1 and Lemma 6.1, the solution obtained is feasible.

We then analyze the time complexity of Algorithm 2 as follows. Let $N = \sum_{i=1}^{L_j} K_i$. The formulation of the BMCGAP takes $O(L_j \cdot C_{max} \cdot d_{max}/c_{min} \cdot |V|)$ time, because there are $L_j = |SFC_j|$ types of items and $\sum_{i=1}^{L_j} K_i \leq \lceil L_j \cdot C_{max}/c_{min} \cdot \max_{v \in V}\{|N_l(v)| + 1\} \rceil = O(L_j \cdot \frac{C_{max}}{c_{min}} \cdot d_{max})$ items. Finding a minimum-cost maximum matching in G_l takes $O(N^3 + |V|^3 + N^2 \cdot |V| + |V|^2 \cdot N)$ time by the Hungarian algorithm, as G_l contains $N + |V|$ nodes.

We claim that the number l of iterations in Algorithm 2 is $O(\log_{\frac{d_{min}}{d_{min^{+1}}}} |\mathcal{I}|)$, which is shown as follows. If an item is not matched in G_l at iteration l, then all of its neighbors in G_l will be matched by other items. Therefore, within each iteration, at most $O(\frac{|\mathcal{I}|}{d_{min}+1})$ items among $|\mathcal{I}|$ items are not matched, where $d_p min$ is the minimum degree of nodes in G_l. Thus, there are $O(\log_{\frac{d_{min}}{d_{min^{+1}}}} |\mathcal{I}|)$ iterations of the proposed algorithm.

Algorithm 2 for the BMCGAP thus takes $O(l \cdot N^3 + l \cdot |V|^3)$ time, where $O(l \cdot N^3 + l \cdot |V|^3) = O(\log_{\frac{d_{min}}{d_{min^{+1}}}} |\mathcal{I}| \cdot (\frac{L_j \cdot C_{max} \cdot d_{max}}{c_{min}})^3 + \log_{\frac{d_{min}}{d_{min^{+1}}}} |\mathcal{I}| \cdot |V|^3) = O((N^3 + |V|^3) \cdot \log_{\frac{d_{min}}{d_{min^{+1}}}} N)$. This is due to the fact that there are no more than N items to be packed to the $|V|$

bins, where $N = |\mathcal{I}| \le \lceil \frac{L_j \cdot C_{max} \cdot d_{max}}{c_{min}} \rceil$, $C_{max} = \max\{C_v \mid v \in V\}$, $c_{min} = \min\{c(f_i) \mid f_i \in SFC_j\}$, $L_j = |SFC_j|$, $d_{min} = \min_{v \in V}\{|N_l(v)|\}$ and $d_{max} = \max_{v \in V}\{|N_l(v)|\}$. In practice, the number of VNF instance backups of each network function is constant, and the values of C_{max}, c_{min}, $|SFC_j|$, d_{min} and d_{max} usually are constants as well. Thus, the running time of Algorithm 2 is $O(|V|^3)$.

Although we only consider that the secondary VNF instances of each primary VNF instance can be placed no more than one-hop neighbor cloudlets from the cloudlet of the primary one, the proposed algorithm is also applicable to the l-hop neighbors of cloudlet v with any fixed l and $2 \le l \le |V| - 1$ directly, the theorem thus follows. □ □

7 PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed algorithms for the service reliability augmentation problem. We also investigate the impact of parameters on the performance of the proposed algorithms.

7.1 Experiment settings

We consider an MEC network $G = (V, E)$ that consists of 100 APs, in which the number of cloudlets is 10% of the network size, and the cloudlets are randomly co-located with some of the APs. Each network topology is generated using the widely adopted approach due to GT-ITM [4]. The computing capacity of each cloudlet ranges from 4,000 to 8,000 MHz [10]. The number $|\mathcal{F}|$ of different types of network functions is set at 30. The computing resource demand of each network function is set from 200 MHz to 400 MHz [2]. For each generated request j, the length $|SFC_j|$ of its service function chain SFC_j is set between 3 and 10, and each network function is randomly drawn from the $|\mathcal{F}|$ types. Each VNF instance in the primary SFC deployed randomly into cloudlets. We assume that its secondary VNF instances can be placed in cloudlets no more than one hop from the primary VNF instance, i.e., $l = 1$. The running time of an algorithm is obtained on a machine with 3.4GHz Intel i7 Quad-core CPU and 16GB RAM. Unless otherwise specified, these parameters will be adopted in the default setting.

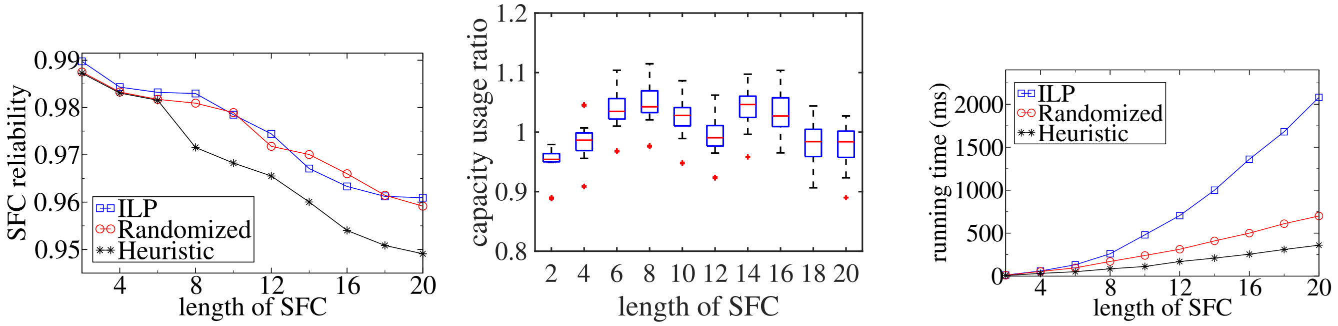
In the following, we evaluate the proposed algorithms, ILP, Algorithm 1 and Algorithm 2. For simplicity, we refer to Algorithm 1 and Algorithm 2 as Randomized, and Heuristic, respectively. For each request with a given length of SFC, 1,000 requests with the same SFC length of the request are randomly generated for each set of experiments. Each value in figures is the mean of the results of these 1,000 trials.

7.2 Performance evaluation

We first evaluate the performance of algorithms Randomized and Heuristic against the exact solution delivered by the ILP for the service reliability augmentation problem, by varying the SFC length of a request from 2 to 20, while fixing the residual computing capacity of each cloudlet at 25%, and the reliability r_i of each network function f_i in the SFC is randomly drawn between 0.8 and 0.9. Fig. 1 illustrates the achieved service function chain reliability, the running times of the three mentioned algorithms, and the ratio of the cloudlet computing capacity usage for algorithm Randomized. It can be seen from Fig. 1(a) that algorithms Randomized and Heuristic can achieve a near optimal service function chain reliability, i.e., the reliabilities delivered by algorithms Randomized and Heuristic are no less than 97.82% and 96.03% of the optimal one, respectively. Notice that, the reliability delivered by algorithm Randomized in some cases is higher than that by ILP, due to allowing violating resource capacity constraints. This has been demonstrated in Fig. 1(b). Fig. 1(b) depicts the average, the minimum, and the maximum computing capacity usage ratio by algorithm Randomized. Fig. 1(c) plots the running time curves of the three mentioned algorithms. It can be seen that the running times of algorithms Randomized and Heuristic are much less than that of ILP, while their solutions are almost comparable to the exact one by the ILP. With the increase on the problem size, the running time of the ILP grows rapidly, and the running time gap between the ILP and the other two algorithms becomes larger and larger. It must be mentioned that the running time of algorithm Heuristic is the least one among the three comparison algorithms for all cases.

We then study the performance of the three mentioned algorithms, by varying the reliability of each network function from 0.6 to 0.9 while keeping other parameters not been changed. Specifically, the reliability of a network function is drawn from intervals [0.55 0.65), [0.65 0.75), [0.75 0.85), and [0.85 0.95], respectively. The results delivered by different algorithm are shown in Fig. 2. It can be seen from Fig. 2(a) that when the network function reliability of each VNF instance increases, the reliability of the service function chain reliability increases at the same time, and the performance gap between the three algorithms becomes smaller. For example, when the average network function reliability is 0.6, algorithm Randomized achieves a service function chain reliability 2.03% less than that by the ILP, and when the average network function reliability is 0.8, algorithm Randomized achieves a service function chain reliability 0.79% less than that by the ILP. Similar performance can be observed for algorithm Heuristic as well. Notice that the service function chain reliability achieved by algorithm Randomized can be higher than that by the ILP due to possible computing resource violation, which is demonstrated in Fig. 2(b). Fig. 2(c) plots the running time curves of the three algorithms, where algorithm ILP takes the longest running time, and algorithm Heuristic takes the least running time.

We finally evaluate the performance of the three mentioned algorithms, by varying the ratio of residual computing capacity of cloudlets to its capacity, while keeping the other parameters unchanged. Fig. 3(a) illustrates the service function chain reliability achieved by different algorithms. It can be seen that when the network has a relatively abundant computing resource, i.e., when there are 50% or 100% of residual computing capacities of each cloudlet, algorithms Randomized and Heuristic can achieve nearly optimal reliability for each request. However, when the residual computing resource in the network becomes less and less, the service function chain reliability decreases. For example, when the network has 50% the residual computing capacity per cloudlet, the three comparison algorithms ILP, Randomized, and Heuristic can deliver solutions with service function chain reliabilities by 98.30%, 97.12%, and 96.42%, respectively; when the computing resource is seriously shortage in network wide, i.e., when there is 1/16 of the residual computing capacity per cloudlet, the service function chain reliabilities achieved by the three algorithms are 66.07%, 62.90%, and

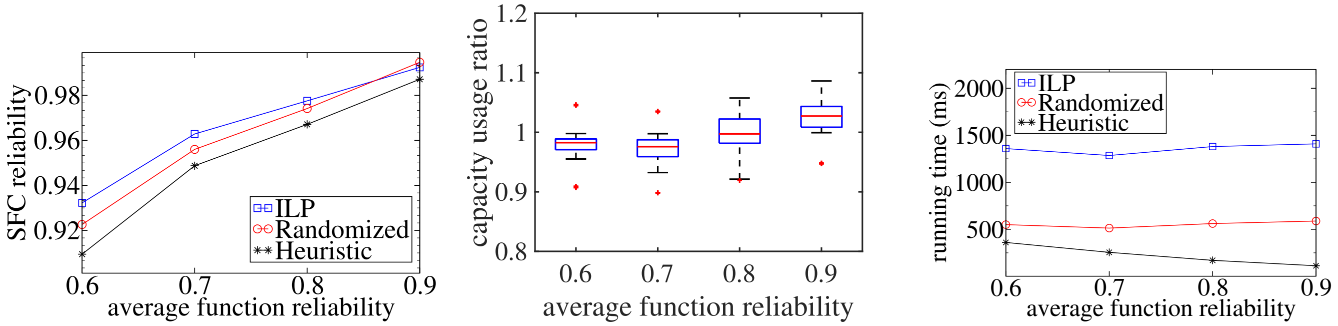


(a) The reliability of a request

(b) The capacity usage ratio of cloudlets by algorithm `Randomized`

(c) The running times of different algorithms

Figure 1: Performance of algorithms `ILP`, `Randomized`, and `Heuristic`, by varying the SFC length of a request from 2 to 20.

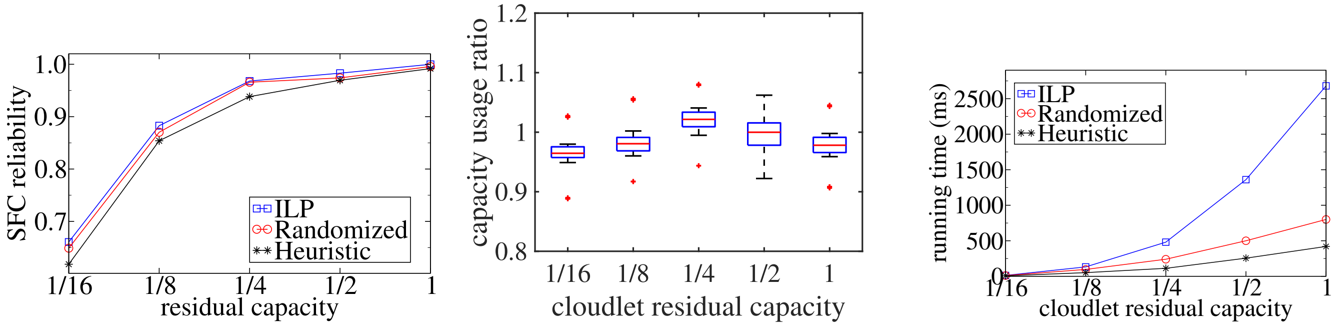


(a) The reliability of a request

(b) The capacity usage ratio of cloudlets by algorithm `Randomized`

(c) The running times of different algorithms

Figure 2: Performance of algorithms `ILP`, `Randomized`, and `Heuristic`, by varying the network function reliability from 0.6 to 0.9.



(a) The reliability of a request

(b) The capacity usage ratio of cloudlets by algorithm `Randomized`

(c) The running times of different algorithms

Figure 3: Performance of algorithms `ILP`, `Randomized`, and `Heuristic` by varying the residual computing capacity of each cloudlet from 1/16 to 1.

60.19%, respectively. The reason behind this is that the amounts of available computing resource in the network determine the number of secondary VNF instances of each primary VNF instance can be instantiated. Fig. 3(b) depicts the computing capacity usage ratio by algorithm `Randomized`, and Fig. 3(c) depicts the running time curves of the three algorithms, from which we can see that with the increase on the residual computing capacity, more secondary VNF instances can be instantiated, and the running times of all the three algorithms increase. Similarly, algorithm ILP takes the longest running time while algorithm Heuristic takes the least running time.

8 CONCLUSIONS

In this paper, we studied a novel reliability augmentation problem for an admitted request with a service function chain and reliability expectation requirements in an MEC network. We enhance the service reliability of the request through placing redundant VNF instances into the cloudlets. We first showed that the problem is NP-hard. We then proposed an integer linear program solution and a randomized algorithm with a good approximation ratio through linear relaxation of the ILP for the problem under the assumption that all the secondary VNF instances must be placed into the cloudlets no more than l hops from the cloudlets of their primary VNF instances. Also, we devised an efficient heuristic algorithm for the problem through reducing the problem to a series of minimum-cost maximum matching problems. We finally evaluated the performance of the proposed algorithms through experimental simulations. Experimental results demonstrate that the proposed algorithms are promising, and their empirical results are superior to their analytical counterparts.

ACKNOWLEDGMENTS

It is acknowledged that the work by Weifa Liang, Yu Ma, and Sid Chi-Kin Chau is supported by the Australian Research Council Discovery Project No. DP200101985. The work by Wenzheng Xu was supported by the National Natural Science Foundation of China (NSFC) with grant number 61602330, Sichuan Science and Technology Program (Grant No. 2018GZDZX0010 and 20. The work by Xiaohua Jia was supported by the Research Grants Council of Hong Kong with Project No. CityU 11214316.

REFERENCES

[1] Saifeddine Aidi, Mohamed Faten Zhani, and Yehia Elkhatib. On improving service chains survivability through efficient backup provisioning. In *Proceedings of CNSM*, IEEE, 2018.

[2] Amazon Web Services, Inc. Amazon EC2 instance configuration. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-ec2-config.html, 2018.

[3] Weiran Ding, Hongfang Yu, and Shouxi Luo. Enhancing the reliability of services in NFV with the cost-efficient redundancy scheme. In *Proceedings of IEEE International Conference on Communication* (ICC), IEEE, 2017.

[4] GT-ITM. http://www.cc.gatech.edu/projects/gtitm/, 2018.

[5] Jingyuan Fan, Chaowen Guan, Yangming Zhao, and Chunming Qiao. Availability-aware mapping of service function chains. In *Proceedings of INFOCOM'17*, IEEE, 2017.

[6] Jingyuan Fan, Meiling Jiang, and Chunming Qiao. Carrier-grade availability-aware mapping of service function chains with on-site backups. In *Proceedings of IWQoS'17*, IEEE, 2017.

[7] Jingyuan Fan, Meiling Jiang, Ori Rottenstreich, Yangming Zhao, Tong Guan, and Ram Ramesh, Sanjukta Das, and Chunming Qiao. A framework for provisioning availability of NFV in data center networks. *IEEE J. of Selected Areas in Communications*, 36(10):2246–2258, 2018.

[8] Bo Han, Vijay Gopalakrishnan, Gnanavelkandan Kathirvel, and Aman Shaikh. On the resiliency of virtual network functions. *IEEE Communications Magazine*, 55:152–157, 2017.

[9] Fujun He, Takehiro Sato, and Eiji Oki. Optimization model for backup resources allocation in middleboxes with importance. *ACM/IEEE Transactions on Networking*, 27(4):1742–1755, 2019.

[10] Hewlett-Packard Development Company. L.P. Servers for enterprise bladeSystem, rack & tower and hyperscale. http://www8.hp.com/us/en/products/servers/, 2015.

[11] Meitian Huang, Weifa Liang, Xiaojun Shen, Yu Ma, and Haibin Kan. Reliability-aware virtualized network function services provisioning in mobile edge computing. *IEEE Transactions on Mobile Computing*, to be published, doi: 10.1109/TMC.2019.2927214, 2019.

[12] Jing Li, Weifa Liang, Meitian Huang, and Xiahua Jia. Providing reliability-aware virtualized network function services for mobile edge computing. In *Proceedings of 39th International Conference on Distributed Computing Systems* (ICDCS'19), IEEE, 2019.

[13] Jing Li, Weifa Liang, Meitian Huang, and Xiaohua Jia. Reliability-aware network service provisioning in mobile edge-cloud networks. *IEEE Transactions on Parallel and Distributed Systems*, 31(7):1545–1558, 2020.

[14] Shouxu Lin, Weifa Liang, and Jing Li. Reliability-aware service function chain provisioning in mobile edge-cloud networks. To appear In *Proceedings of 29th International Conference on Computer Communications and Networks*, August 3 – August 6, Hawaii, USA, IEEE, 2020.

[15] Yu Ma, Weifa Liang, Jie Wu, and Zichuan Xu. Throughput maximization of NFV-enabled multicasting in mobile edge cloud networks. *IEEE Transactions on Parallel and Distributed Systems*, 31(2):394–407, 2020.

[16] Robert M. Nauss. Solving the generalized assignment problem: an optimizing and heuristic approach. *INFORMS Journal of Computing*, 15(3):249–266, 2003.

[17] Long Qu, Chadi Assi, Khaled Shaban, and Maurice J. Khabbaz. A reliability-aware network service chain provisioning with delay guarantees in NFV-enabled enterprise datacenter networks. *IEEE Transactions on Network Service Managements*, 14(3):554–568, 2017.

[18] Long Qu, Maurice Khabbaz, and Chadi Assi. Reliability-aware network service chaining in carrier-grade softwarized networks. *IEEE J. Sec. Areas Commun.*, 36(3):558–573, 2018.

[19] Prabhakar Raghavan, and Clark D. Tompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374, 1987.