Request Reliability Augmentation with Service Function Chain Requirements in Mobile Edge Computing

Weifa Liang, *Senior Member, IEEE,* Yu Ma, Wenzheng Xu, *Member, IEEE,* Zichuan Xu, *Member, IEEE,* Xiaohua Jia *Fellow, IEEE,* and Wanlei Zhou, *Senior Member, IEEE*

APPENDIX

Theorem 1. The service reliability augmentation problem for an admitted request with a SFC and a reliability expectation requirements in an MEC network $G = (V, E)$ is NP-hard.

Proof We reduce the minimum-cost generalized assignment problem (GAP) [22] 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 service reliability augmentation 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 to pack as many items as possible 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 [22].

We 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 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, $u \in N_l^+(v) = V$ if the primary VNF instance of f_i is placed at cloudlet v, 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. □

- *W. Liang and X. Jia are with Department of Computer Science, City University of Hong Kong, 83 Tat Chee Ave., Kowloon, Hong Kong. E-mails: weifa.liang@cityu.edu.hk, and csjia@cityu.edu.hk*
- *Y. Ma is with Research School of Computer Science, The Australian National University, Canberra, ACT 2601, Australia. E-mail: yu.ma@anu.edu.au*
- *W. Xu is with School of Computer Science, Sichuan University, Chengdu, 510006, P. R. China. wenzheng.xu@scu.edu.cn*
- *Z. Xu is with School of Software, Dalian University of Technology, Dalian, 116621, P. R. China. z.xu@dult.edu.cn*
- *Wanlei Zhou is with the Institute of Data Science, City University of Macau, Macao SAR, China. E-mail: wlZhou@cityu.mo.*

Theorem 2. Given a request j with SFC_j in G, let $G_j = (N_j \cup \{s_j, t_j\}, A_j; \omega)$ be the auxiliary graph constructed for the admission of request j. If there is not any directed path in G_j from s_j to t_j, then request j is not admissible due to lack of computing resource to accommodate the VNF instances of its SFC_j. Otherwise, a shortest path in G_j from s_j to t_j in terms of the defined edge weight function $\omega(\cdot, \cdot)$ corresponds a feasible VNF instance placement of network functions in SFC_j for request j, and the reliability achieved of this placement is the maximum one. The algorithm takes $O(|V|^2 \cdot L_j)$ time, where $L_j = |SFC_j|$.

Proof We first show whether request j is admissible or not. Let $CC(s_j)$ be the set of nodes in G_j in which each node is reachable from node s_j, and let $RR(t_j)$ be the set of nodes in G_j that node t_j is reachable from any node in $RR(t_j)$. It can be seen that if $CC(s_j) \cap RR(t_j) = \emptyset$, request j is inadmissible due to lack of computing resource in the MEC network; otherwise, there exists at least one node $u \in CC(s_j) \cap RR(t_j)$, from which a directed path in G_j from s_j to t_j can be constructed, which consists of two segments: one is from s_j to u as node u is reachable from s_j, and the other is from u to t_j as $u \in RR(t_j)$. Since G_j is a DAG, there is no directed cycles in G_j, and the directed path must be a simple path, which means that the path delivers

a feasible solution for the primary VNF instance placement of request j in the order to traverse along the VNF instances exactly as the order in SFC_j, following the construction of G_j.

The rest is to show that a shortest path P in G_j from s_j to t_j corresponds the primary VNF instance placement of SFC_j with the maximum reliability. Let $s_j, v_1, v_2, \ldots, v_{L_j}, t_j$ be the node sequence of P. The length $l(P)$ of P in G_j is

$$\begin{aligned} l(P) &= \omega(s_j, v_1) + \sum_{l=1}^{L_j-1} \omega(v_l, v_{l+1}) + \omega(v_{L_j}, t_j) \\ &= -\sum_{l=1}^{L_j} \log r_l + (-\log 1), \quad \text{as } \omega(s_j, v_1) = -\log r_1 \\ &= -\sum_{l=1}^{L_j} \log r_l. \end{aligned} \tag{1}$$

Since the length $l(P)$ $(= -\sum_{l=1}^{L_j} \log r_l)$ of P is the minimum one, the value of $-l(P)$ is the maximum one. The reliability of request j, in terms of its primary VNF instance chain placement derived from P, thus is the maximum one among all directed paths in G_j from s_j to t_j, which is $2^{-l(P)} = 2^{\sum_{l=1}^{L_j} \log r_l} = \Pi_{l=1}^{L_j} r_i$. Note that we here simplify the assumption that the reliability of any VNF instance of $f_i \in \mathcal{F}$ at any cloudlet is the same, i.e., r_i, this assumption has been widely adopted in literature [8], [7], [11], [13]. In fact, the proposed framework is still applicable to the case where the VNF instances of f_i at different cloudlets have different reliabilities through the weight assignment of their corresponding directed edges in G_j.

The running time of the proposed algorithm, `Algorithm 1`, is analyzed as follows. Given request j, the construction of G_j takes $O(|V|^2 \cdot L_j)$ time. Finding a shortest path in G_j from s_j to t_j takes $O(|N_j| + |A_j|) = O(|V|^2 \cdot L_j)$ time, as G_j is a DAG, and it takes a linear time to find a shortest path in any DAG. □

Theorem 3. Given an MEC network $G(V, E)$ and a request j with SFC_j and reliability expectation ρ_j, there is a randomized algorithm, `Algorithm 2`, 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 number of secondary VNF instances for function $f_i \in SFC_j$, Λ is a constant strictly greater than 2, P^* is the optimal reliability of request j in G, and Λ is a constant defined in Eq. (2).

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.

$$\begin{aligned} \Lambda = \max\{ &\max\{c(f_i, k_i, *) \mid I_{k_i} \in \mathcal{I}\}, \max\{C'_u \mid u \in V\}, \\ &-\log \rho_j, \max\{c(f_i) \mid f_i \in SFC_j\}\}. \end{aligned} \tag{2}$$

Let $\widetilde{OPT}$ be the optimal solution of the linear program (LP). Clearly, the value of $\widetilde{OPT}$ 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,

$$\mathbb{E}[\sum_{I_{k_i} \in \mathcal{I}} y_{i,k_i,u}] = \sum_{I_{k_i} \in \mathcal{I}} \frac{c(f_i, k_i, u) \cdot \tilde{x}_{i,k_i,u}}{\Lambda} = \frac{\widetilde{OPT}}{\Lambda}. \tag{3}$$

Let $\mu = \mathbb{E}[\sum_{I_{k_i} \in \mathcal{I}} c(f_i, k_i, u) \cdot \tilde{x}_{i,k_i,u}] = \widetilde{OPT}$. Following the Chernoff bound in Lemma 3 (i), we have

$$\begin{aligned} &\mathbf{Pr}[\sum_{I_{k_i} \in \mathcal{I},\ f_i \in N_{f,v},\ \&\ u \in N_l^+(v)} c(f_i, k_i, u) \cdot x_{i,k_i,u} \geq (1+\beta) \cdot OPT] \\ &\leq \mathbf{Pr}[\sum_{I_{k_i} \in \mathcal{I},\ f_i \in N_{f,v},\ \&\ u \in N_l^+(v)} c(f_i, k_i, u) \cdot x_{i,k_i,u} \geq (1+\beta) \cdot \widetilde{OPT}], \\ &\qquad \text{since } \widetilde{OPT} \leq OPT \end{aligned} \tag{4}$$

$$\begin{aligned} &= \mathbf{Pr}[\sum_{I_{k_i} \in \mathcal{I},\ f_i \in N_{f,v},\ \&\ u \in N_l^+(v)} y_{i,k_i,u} \geq (1+\beta) \cdot \frac{\widetilde{OPT}}{\Lambda}] \\ &\leq exp(\frac{-\beta^2 \cdot \mu}{2+\beta}) \quad \text{for all } \beta > 0. \end{aligned} \tag{5}$$

We then assume that

$$exp(\frac{-\beta^2 \cdot \frac{\mu}{\Lambda}}{2+\beta}) \leq \frac{1}{N}, \tag{6}$$

$$\beta > 0. \tag{7}$$

When $0 < \beta \leq 1$, Inequality (6) is transformed as follows.

$$exp(\frac{-\beta^2 \cdot \mu}{3\Lambda}) \leq \frac{1}{N}, \tag{8}$$

when $N = |\mathcal{I}|$ is sufficiently large, the solution of Ineq. (8) is

$$\beta \geq \sqrt{\frac{3\Lambda \ln N}{\mu}} = \sqrt{\frac{3\Lambda \ln N}{\widetilde{OPT}}} \geq \sqrt{\frac{3\Lambda \ln N}{OPT}}. \tag{9}$$

Since $\beta \leq 1$, we must have $OPT \geq 3\Lambda \ln N$. Thus, the optimal reliability P^* of request j is at least

$$P^* \geq 2^{-OPT} = (\frac{1}{2^{\ln N}})^{3\Lambda} = \frac{1}{N^{3\Lambda/\log e}}. \tag{10}$$

The approximation ratio of the randomized algorithm, `Algorithm 2`, 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 (8), 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 (8), 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

$$\frac{2^{-A}}{P^*} \geq \frac{P^{*\frac{2}{\Lambda}}}{P^*} = P^{*(\frac{2}{\Lambda}-1)} = \frac{1}{P^{*(1-\frac{2}{\Lambda})}}. \quad (11)$$

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,

$$\mu_1 = \mathbb{E}[\sum_{I_{k_i} \in \mathcal{I},\ f_i \in N_{f,v},\ \&\ u \in N_l^+(v)} z_{i,k_i,u}]$$
$$= \sum_{I_{k_i} \in \mathcal{I},\ f_i \in N_{f,v},\ \&\ u \in N_l^+(v)} \frac{c(f_i) \cdot \tilde{x}_{i,k_i,u}}{\Lambda} = \frac{\tilde{C}'_u}{\Lambda}, \quad (12)$$

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

$$\mathbf{Pr}[\bigvee_{v \in V} \sum_{I_{k_i} \in \mathcal{I},\ f_i \in N_{f,v},\ \&\ u \in N_l(v) \cup \{v\}} c(f_i) \cdot x_{i,k_i,u} \geq (1+\beta_1) \cdot C'_u]$$
$$\leq \mathbf{Pr}[\bigvee_{v \in V} \sum_{I_{k_i} \in \mathcal{I},\ f_i \in N_{f,v},\ \&\ u \in N_l^+(v)} c(f_i) \cdot x_{i,k_i,u} \geq (1+\beta_1) \cdot C'_u]$$
since $\tilde{C}'_u \leq C'_u$ (13)
$$= \sum_{v \in V} \mathbf{Pr}[\sum_{I_{k_i} \in \mathcal{I},\ f_i \in N_{f,v},\ \&\ u \in N_l^+(v)} z_{i,k_i,u} \geq (1+\beta_1) \cdot \frac{\tilde{C}'_u}{\Lambda}]$$
$$\leq |V| \cdot exp(-\frac{\beta_1^2 \cdot \mu_1}{2+\beta_1}), \quad \text{by the Chernoff bound.} \quad (14)$$

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

$$exp(-\frac{\beta_1^2 \cdot \mu_1}{2+\beta_1}) \leq \frac{1}{|V|^2}. \quad (15)$$

Then,

$$\beta_1 \geq \sqrt{\frac{6 \ln |V|}{\mu_1}} = \sqrt{\frac{6\Lambda \ln |V|}{\tilde{C}'_u}} \geq \sqrt{\frac{6\Lambda \ln |V|}{C'_u}} \quad \text{since } \tilde{C}'_u \leq C'_u. \quad (16)$$

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

$$\min\{C'_u \mid u \in V\} \geq 6\Lambda \ln |V|. \quad (17)$$

We then have

$$\mathbf{Pr}[\bigvee_{v \in V} \sum_{I_{k_i} \in \mathcal{I},\ f_i \in N_{f,v},\ \&\ u \in N_l(v) \cup \{v\}} c(f_i) \cdot x_{i,k_i,u} \geq (1+\beta_1) \cdot C'_u]$$
$$\leq \mathbf{Pr}[\bigvee_{v \in V} \sum_{I_{k_i} \in \mathcal{I},\ f_i \in N_{f,v},\ \&\ u \in N_l^+(v)} c(f_i) \cdot x_{i,k_i,u} \geq (1+\beta_1) \cdot \tilde{C}'_u]$$
$$\leq |V| \cdot exp(-\frac{\beta_1^2 \cdot \mu_1}{2+\beta_1}) \quad \text{by (15)} \quad (18)$$
$$\leq |V| \cdot \frac{1}{|V|^2} = \frac{1}{|V|}. \quad (19)$$

The theorem then follows. □

Theorem 4. 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 3`, for the service reliability augmentation problem of an admitted request j, under the assumption that all the 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 a fixed integer with $1 \leq l \leq |V|-1$. The the time complexity of `Algorithm 3` 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 3`, 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 secondary 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 1 and Lemma 4, the solution obtained is feasible.

We then analyze the time complexity of `Algorithm 3` as follows. Let $N = \sum_{i=1}^{L_j} K_i$. The formulation of the BMCGAP takes $O(N \cdot |V|) = O(\sum_{i=1}^{L_j} K_i \cdot |V|) = O(\sum_{i=1}^{L_j} \frac{C_{max} \cdot d_{max}}{c_{min}} \cdot |V|) = O(L_j \cdot C_{max} \cdot d_{max}/c_{min} \cdot |V|)$ 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 $O((N+|V|)^3) = O(N^3+|V|^3+N^2\cdot|V|+|V|^2\cdot N)$ time by the Hungarian algorithm, as G_l contains $N+|V|$ nodes.

We claim that the number of iterations l in `Algorithm` 3 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_pmin 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, i.e., $O(\log_{\frac{d_{min}}{d_{min}+1}} |\mathcal{I}|)$ bipartite graphs will be constructed in the algorithm.

`Algorithm` 3 for the BMCGAP thus takes $O(l\cdot N^3 + l\cdot|V|^3 + l\cdot N^2\cdot|V| + l\cdot|V|^2\cdot N) = O(l\cdot N^3 + l\cdot|V|^3)$ time, where $O(l\cdot N^3 + l\cdot|V|^3) = O(\log_{\frac{d_{min}}{d_{min}+1}} |\mathcal{I}| \cdot (\frac{L_j\cdot C_{max}\cdot d_{max}}{c_{min}})^3 + \log_{\frac{d_{min}}{d_{min}+1}} |\mathcal{I}|\cdot|V|^3) = O(\log_{\frac{d_{min}}{d_{min}+1}} N \cdot (L_j\cdot C_{max}\cdot d_{max}/c_{min})^3 + \log_{\frac{d_{min}}{d_{min}+1}} N\cdot|V|^3) = O((N^3+|V|^3)\cdot\log_{\frac{d_{min}}{d_{min}+1}} N)$. This is due to the fact that there are no more than N items to be packed to the $|V|$ bins, where $N = |\mathcal{I}| \leq \lceil \frac{L_j\cdot C_{max}\cdot d_{max}}{c_{min}} \rceil$, $C_{max} = \max\{C_v \mid v \in V\}$, $c_{min} = \min\{c(f_i) \mid f_i \in SFC_j\}$, $L_j = |SFC_j|$, $d_{min} = \min_{v\in V}\{|N_l(v)| \mid v \in V\}$ and $d_{max} = \max_{v\in V}\{|N_l(v)| \mid v \in V\}$. 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` 3 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 $2 \leq l \leq |V|-1$ directly, the theorem thus follows. □

Theorem 5. Given an MEC network $G(V,E)$ and a group Q of admitted requests with each request $q_j \in Q$ with a SFC_j and a reliability expectation ρ_j, each cloudlet $v \in V$ has computing capacity C_v, there is an algorithm, `Algorithm` 4, for the service reliability augmentation problem for a set Q of admitted requests, under the assumption that all secondary VNF instances of each primary VNF instance must be placed in the cloudlets in $N_l(v)$ if the primary VNF instance is placed in cloudlet $v \in V$ with a fixed integer l and $1 \leq l \leq |V|-1$. `Algorithm` 4 takes $O(|Q|\cdot(N^3+|V|^3)\cdot\log_{\frac{d_{min}}{d_{min}+1}} N)$, where $N = \max_{1\leq j\leq|Q|}\{\lceil\sum_{i=1}^{L_j} K_{j,i}\rceil \leq \lceil\frac{L_j\cdot C_{max}\cdot d_{max}}{c_{min}}\rceil\}$, $d_{min} = \min\{d_v \mid v \in V\}$, $d_{max} = \max\{d_v \mid v \in V\}$, $L_{max} = \max\{L_j \mid q_j \in Q, \& \ L_j = |SFC_j|\}$, $C_{max} = \max\{C_v \mid v \in V\}$, $c_{min} = \min\{c(f_i) \mid f_i \in \mathcal{F}\}$, $d_{max} = \max_{v\in V}\{|N_l(v)| \mid v \in V\}$.

Proof: The proof body is almost identical to the one in the proof of Theorem 4, omitted. □

REFERENCES

[1] S. Aidi, M. F. Zhani, and Yehia Elkhatib. On improving service chains survivability through efficient backup provisioning. *Proc. of International Conference on Network and Service Management (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] D. Chemodanov, P. Calyam, and F. Esposito. A near optimal reliable composition approach for geo-distributed latency-sensitive service chains. *Proc. of IEEE Conference on Computer Communications*(INFOCOM'19), IEEE, pp.1792–1800, 2019.

[4] W. Ding, H. Yu, and S. Luo. Enhancing the reliability of services in NFV with the cost-efficient redundancy scheme. *Proc. of International Conference on Communication*(ICC'17), IEEE, 2017.

[5] H. Feng, J. Llorca, A. M. Tulino, D. Raz, and A. F. Molish. Approximation algorithms for the NFV service distribution problem. *Proc. of INFOCOM*, IEEE, 2017.

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

[7] J. Fan, C. Guan, Y. Zhao, and C. Qiao. Availability-aware mapping of service function chains. *Proc. of INFOCOM'17*, IEEE, 2017.

[8] J. Fan, M. Jiang, and C. Qiao. Carrier-grade availability-aware mapping of service function chains with on-site backups. *Proc. of IWQoS'17*, IEEE, 2017.

[9] J. Fan, M. Jiang, O. Rottenstreich, Y. Zhao, T. Guan, R. Ramesh, S. Das, and C. Qiao. A framework for provisioning availability of NFV in data center networks. *IEEE J. of Selected Areas in Communications*, Vol.36, No.10, pp. 2246–2258, 2018.

[10] B. Han, V. Gopalakrishnan, G. Kathirvel, and A. Shaikh. On the resiliency of virtual network functions. *IEEE Communications Magazine*, Vol. 55, pp. 152–157, 2017.

[11] F. He, T. Sato, and E. Oki. Optimization model for backup resources allocation in middleboxes with importance. *ACM/IEEE Trans. on Networking*, Vol.27, No.4, pp. 1742–1755, 2019.

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

[13] M. Huang, W. Liang, X. Shen, Y. Ma, and H. Kan. Reliability-aware virtualized network function services provisioning in mobile edge computing. *IEEE Transactions on Mobile Computing*, Vol. 19, No.11, pp. 2699–2713, 2020.

[14] M. Jia, W. Liang, and Z. Xu. QoS-aware task offloading in distributed cloudlets with virtual network function services. *Proc. of MSWiM'17*, ACM, 2017.

[15] J. Li, W. Liang, M. Huang, and X. Jia. Providing reliability-aware virtualized network function services for mobile edge computing. *Proc. of 39th International Conf. on Distributed Computing Systems* (ICDCS'19), July, IEEE, 2019.

[16] J. Li, W. Liang, M. Huang, and X. Jia. Reliability-aware network service provisioning in mobile edge-cloud networks. *IEEE Transactions on Parallel and Distributed Systems*, Vol.31, No.7, pp. 1545–1558, 2020.

[17] J. Li, W. Liang, and Y. Ma. Robust service provisioning with service function chain requirements in mobile edge computing. *IEEE Transactions on Network and Service Management*, to be published, 2021, DOI: 10.1109/TNSM.2021.3062650

[18] W. Liang, Y. Ma, W. Xu, X. Jia, and S. Chau. Reliability augmentation of requests with service function chain requirements in mobile edge-cloud networks *Proceedings of 49th Intl Conf on Parallel Processing* (ICPP'20), ACM, 2020.

[19] S. Lin, W. Liang, and J. Li. Reliability-aware service function chain provisioning in mobile edge-cloud networks. *Proceedings of 29th International Conference on Computer Communications and Networks* (ICCCN'20), IEEE, 2020.

[20] Y. Ma, W. Liang, J. Wu, and Z. Xu. Throughput maximization of NFV-enabled multicasting in mobile edge cloud networks. *IEEE Transactions on Parallel and Distributed Systems*, Vol.31, No.2, pp.394–407, 2020.

[21] Y. Mao, C. You, J. Zhang, K, Huang, and K. Letaief. A survey on mobile edge computing: the communication perspective. *IEEE Commun. Surv. Tutor.*, vol. 19, pp. 2322 – 2358, 2017.

[22] R. M. Nauss. Solving the generalized assignment problem: an optimizing and heuristic approach. *INFORMS Journal of Computing*, Vol.15, No.3, pp.249–266, 2003.

[23] L. Qu, C. Assi, K. Shaban, and M. J. Khannaz. A reliability-aware network service chain provisioning with delay guarantees in NFV-enabled enterprise datacenter networks. *IEEE Transactions on Network Service Managements*, Vol.14, No.3, pp.554 – 568, 2017.

[24] L. Qu, K. Shaban, and A. Assi. Reliability-aware network service chaining in carrier-grade softwarized networks. *IEEE J. Sec. Areas Commun.*, Vol.36, No.3, pp.558 – 573, 2018.
[25] P. Raghavan and C. D. Thompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, Vol.7, No.4, pp.365–374, 1987.
[26] B. G. Rodriguez-Santana, A. M. Viveros, B. E. Carvajal-Gamez, and D. C. Trejo-Osorio. Mobile computation offloading architecture for mobile augmented reality, case study: visualization of cetacean skeleton. *International Journal of Advanced Computer Science & Applications*, vol. 1, no. 7, pp. 665 – 671, Jun. 2016.
[27] X. Shang, Y. Huang, Z. Liu, and Y. Yang. Reducing the service function chain backup cost over the edge and cloud by a self-adapting scheme. *Proc. of IEEE Conference on Computer Communications*(INFOCOM'20), IEEE, pp.2096–2105, 2020.
[28] Z. Xu, W. Liang, M. Jia, M. Huang, and G. Mao. Task offloading with network function services in a mobile edge-cloud network. *IEEE Transactions on Mobile Computing*, vol. 18, no. 11, pp. 2072 – 2685, 2019.
[29] S. Yang, F. Li, R. Yahyappour, X. Fu. Delay-sensitive and availability-aware virtual network function scheduling for NFV. To appear in *IEEE Transactions on Service Computing*, 2019.
[30] R. Yu, G. Xue, and X. Zhang. QoS-aware and reliable traffic steering for service function chaining in mobile networks. *IEEE Journal on Selected Areas in Communications*, Vol.35, No.11, pp.2522–2531, 2017.