Received: 14 February 2020 | Accepted: 14 February 2020

DOI: 10.1002/net.21935

SPECIAL ISSUE ARTICLE

WILEY

Robust network function virtualization

Tachun Lin[1] | Zhili Zhou[2]

[1]Department of Computer Science and Information Systems, Bradley University, Peoria, Illinois
[2]Grubhub, Chicago, Illinois

Correspondence
Tachun Lin, Department of Computer Science and Information Systems, Bradley University, Peoria, IL 61625, USA.
Email: djlin@bradley.edu

Abstract

Network function virtualization (NFV) enables on-demand network function (NF) deployment providing agile and dynamic network services. Through an evaluation metric that quantifies the minimal reliability among all NFs for all demands, service providers and operators may better facilitate flexible NF service recovery and migration, thus offering higher service reliability. In this paper, we present evaluation metrics on NFV reliability and solution approaches to solve robust NFV under random NF-enabled node failure(s). We demonstrate how to construct an auxiliary NF-enabled network and its mapping onto the physical substrate network. With the constructed NF-enabled network, we develop pseudo-polynomial algorithms to solve the robust NF and SFC $s-t$ path problems: subproblems of robust NFV. We also present approximation algorithms for robust NFV with the SFC-Fork as the NF forwarding graph. Furthermore, we propose exact solution approaches via mixed-integer linear programming under the general setting. Computational results show that our proposed solution approaches are capable of managing robust NFV in a large-size network.

KEYWORDS

approximation algorithm, cross-layer network, network function virtualization, quality-of-service, robustness of network function service, service function chaining

1 | INTRODUCTION

The development of 5G networks targets to deliver ultra-reliable and super low latency communication [6, 43], which supports dynamic requests over large-scale cross-domain networks. Through network function virtualization (NFV), a 5G-enabling technique, network functions (NFs) are decoupled from costly proprietary networking hardware and are realized through their software implementation of virtual network functions (VNFs) running on industry-standard commercial off-the-shelf hardware [3, 29]. Radio signal processing [44] and mobile/optical networking [37] are also applying the NFV and deploy VNFs on NF-enabled physical infrastructures, such as virtual machines and containers, and provide on-demand NF services [24].

NF service providers provision, manage, and orchestrate VNFs with NFV management and orchestration architectures (MANO) [22] for end users which request a sequence of NFs called "service function chaining" (SFC). An instance of SFC is (firewall → intrusion prevention system → load balancer). We let "non-chained" NF requests denote the NF requests without a specified sequence. With NFV, network operators allocate and reallocate VNF instances and route network traffic between service functions. Hence, NFV does not only provide more flexibility but also shorten the enabling time of new NF services [22]. To realize end-user demands with NF requests, the NF provisioning problem, which determines the physical infrastructure to deploy VNF instances to fulfill NF requests, arises. We illustrate an instance of NF provisioning problem in Figure 1.

NFV MANO also supports NF recovery and migration, the major approaches to guarantee continuity, resilience, and security of NF services [20, 37, 50]. When a VNF instance is not reachable [21], MANO initiates the fail-over to other available NF instances and automatically recovers NF services, or instantiates new VNFs [52]. Meanwhile, dynamic and flexible VNF

Tachun Lin and Zhili Zhou contributed equally to this study.

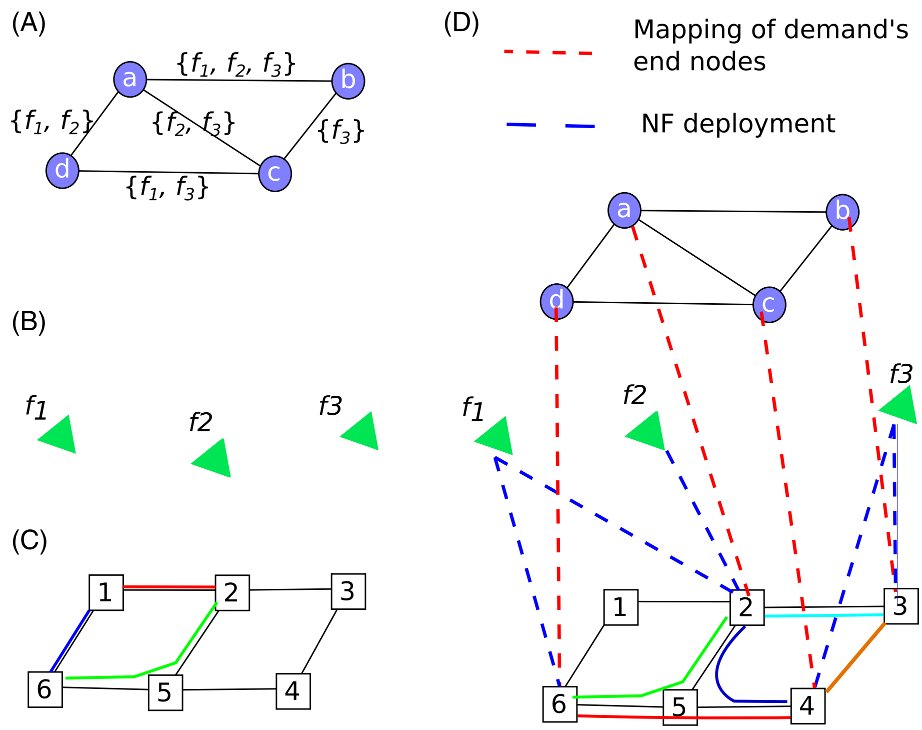

FIGURE 1 NF provisioning with NFs deployment on NF-enabled nodes. A, Demands with NF requests. B, NFs. C, Physical network. D, NF deployment and routing [Color figure can be viewed at wileyonlinelibrary.com]

migration also reduces power consumption and the burden on hardware capacity [37]. To support NF recovery and migration, reserving physical resources should also be considered in the NF provisioning problem.

While the above studies provide valuable insights from different aspects of NF services, they cannot be used to quantify system capability and reliability to support NF recovery and migration [19, 30, 31] as well as seamless NFV state transitions [41, 51] under component failure(s). Motivated by the objective of providing ultra-reliable 5G services, we studied the robust NF provision problem in [35] which takes into account the uncertain failures of NF-enabled nodes from the network operators' perspective. Robust evaluation metrics proposed in [35] on robust VNF provisioning aim to provide VNF managers/orchestrators a way to evaluate the strategies to instantiate VNFs on available NF-enabled nodes (NF resource pools) based on the information of the physical infrastructure and resource utilization.

This study addresses robust NF provisioning and related network design and routing problems. The robust NF provisioning problem determines the location of VNF instance deployment, and possible NF request fulfillment is determined with a robust evaluation metric as the objective to handle random NF-enabled node failures. Extended from [35], we study three sets of network design and routing problems for both considering and not considering NF-enabled node failure: (1) the minimal weighted SFC $s-t$ path problem, which determines the minimal weighted path for SFC requests visiting all required NFs in sequence through NF-enabled nodes; (2) VNF provisioning with SFC-Fork as the forwarding graph structure, which determines deployment locations of VNF instances realizing all SFC requests; and (3) VNF provisioning with general NF forwarding graph. The first set of problems is a fundamental problem in NFV which helps establish the end-to-end route to fulfill NF requests. The second set has SFC-Fork as the NF forwarding graph, which is the commonly established forwarding graph in NFV 5G implementation (see [15, 32, 38, 42]). To address these problems, we construct an auxiliary NF-enabled network that serves as an intermediate layer between the NF forwarding graph and the physical substrate network and provides all possible connections among NF-enabled nodes. We present pseudo-polynomial algorithms for the NF and SFC $s-t$ path problems and approximation algorithms for the NF provisioning problem with SFC-Fork. We also validate our proposed solution approaches with computational results over small and large scale national-wide physical networks.

We highlight our contributions in this paper as follows.

1. We propose robust evaluation metrics [35] on robust VNF provisioning to provide VNF managers/orchestrators a way to evaluate strategies in instantiating VNFs on available NF-enabled nodes (in NF resource pools) based on the physical infrastructure and resource utilization.
2. We construct multilayer graphs and establish their corresponding mapping relationships, which provide network structures to solve the NFV design problems.

3. We provide pseudo-polynomial algorithms for NF and SFC $s-t$ path problems in both deterministic and robust settings.
4. We demonstrate the existence of bi-factor approximation algorithms on NF provisioning with SFC-Fork and propose corresponding algorithms.
5. We propose a two-step parameterized path reduction technique in approximation algorithm design to manage branching structures in tree networks.
6. With the insight obtained from the approximation algorithm for the optimal NF provisioning problem, we develop an approximation algorithm for the robust NF provisioning problem.

In short, the robust NF provisioning problems address the problem of sequential location selection/resource allocation and the routing through selected locations, which add new variants and dimensions in the traditional location and location-routing problems. Our proposed evaluation metrics and solution approaches serve the purpose of dealing with these new variants.

The rest of the paper is organized as follows. We first present the related works, problems, and their solution approaches in Section 2. Especially, we summarize the approximation algorithms for k-level facility location problems and their robust/fault-tolerant relatives. In Section 3, we review the evaluation metric for NF services, the robust NF-service evaluation metric, for non-chained NFs and SFCs and define the robust NF provisioning problem and related subproblems. We present corresponding solution approaches in Section 4, where an auxiliary NF-enabled network is constructed. We introduce the (robust) SFC path algorithms and the mixed-integer programming formulations to solve robust NF provisioning for non-chained NF and SFC requests, respectively. The experiment setting and computational results to validate our proposed approaches are given in Section 5. We also demonstrate the lower bound benchmark of the robust NF-service evaluation metrics, followed by the conclusions and future research directions in Section 6. We also include in the Appendix supporting theorems and proofs.

2 | LITERATURE REVIEW

2.1 | NFV techniques

We review in this section works on NFV resource allocations and existing solution approaches, and we focus on the approximation algorithms for facility location problems and relevant location routing problems. Related topics have been reviewed in survey papers, such as NFV architectures [8], mobile applications [40], and NF deployment/provisioning related resource allocation [39, 53]. Most recent works on NFV MANO focus on NFV instantiation [12, 56], orchestration [54], management [18], and scheduling [5]. Newly developed technologies are capable to support NFV in various telecommunication systems.

2.2 | Related problems and solution approaches

Most resource allocation problems for NF deployment/provisioning [39] are under the setting that given a physical substrate network (an available NFVI), the set of NF-enabled nodes is a subset of physical nodes and the end-to-end NF demands are established/realized through paths in the physical substrate network. The NF deployment problem determines locations and copies of VNF instances deployed, and generates end-to-end routes (static/dynamic) for NF requests. On top of the NF deployment problem, the NF provisioning problem further estimates physical resources considering also the quality-of-service (QoS). Hence, NF deployment and provisioning problems belong to location-routing problems.

SFC route generation [47] and NF forwarding graphs embedding [28] are the two approaches to manage SFC requests either individually or jointly. Hence, even with the simplest case, the minimal weighted SFC path problem is different from the shortest path problem, which requires visiting VNF instantiated physical nodes in the desired sequence. Cohen et al. [17] decomposes non-chained VNF deployment into two stages: (1) VNF instance deployment via the facility location problem, and (2) VNF instance assignment for NF service requests via the assignment problem, which allows NF service requests to be fulfilled by splittable flows. They provide a mixed integer linear programming (MILP) formulation for each stage with the objective to minimize the system-wide operation costs and build an upper-bounded heuristic algorithm. Bari et al. [7] study SFC deployment and present a MILP formulation which realizes SFC requests via multiple virtual network embeddings [16]. A heuristic algorithm based on a multistage directed graph and the Viterbi algorithm is proposed in [7], which takes each SFC as a virtual network and maps each virtual network onto the given physical substrate network. Rost and Schmid [46] study multiple SFCs (forwarding graph) embedding and propose a polynomial-time approximation algorithm based on random routing techniques with linear programming (LP) relaxation. Even et al. [25] provide a randomized approximation algorithm leveraging multicommodity flows for path computation and function placement. Sallam et al. [47] focus on the SFC counterpart of standard graph theory problems, such as the minimal weighted SFC path and the SFC maximal flow. The proposed pseudo-polynomial algorithms solve the minimal weighted SFC paths on a transferred network graph and a special case of SFC maximal flow problem.

With the capacity limitation on the physical substrate network and single-path realization of end-user demands, Lin et al. [36] deploy VNF instances in an optical backbone network and formulate the problem as a unit flow multicommodity flow problem. The resource competition between NF instantiation and demand routing is captured using a game theory setting, which reveals that a system-wide optimal solution may not be preferable for network service providers and individual end-users. D'Oro et al. [18] take into account the selfishness and competitiveness of end-users' behavior and formulate an atomic weighted congestion game for SFC routing. It proposes a polynomial-time algorithm to achieve Nash equilibrium with a bounded price of anarchy. Besides planning and operation costs, QoS is another key evaluation metric for network services, especially in a 5G environment. One of the 5G-PPP research projects funded by the European Union, 5G-NORMA [1], provides QoS requirements on NF service during 5G implementations considering deployment failure due to lack of infrastructure resources. Fan et al. [26] demonstrate that controlled redundancy provides extra protection and recovery capability for network services when a physical failure occurs. It develops an online heuristic algorithm minimizing physical resource consumption and guaranteeing service reliability with backup resources. Qu et al. [45] study the reliability-aware NF provisioning problem and propos an MILP formulation and greedy based heuristic algorithm, in which extra backup VNF instances for NF requests are deployed. From the operators' perspective on managing the QoS of NF services, in this study, we fill in the gap with the study of the QoS-controlled NF provisioning problem. We first propose a new evaluation metric on NF-service reliability under the worst-case scenario, followed by studying *robust NF provisioning* under the failure uncertainty of NF-enabled node. As mentioned earlier, two subproblems are involved in the NF provisioning problem: (1) the minimal-weighted SFC path problem: different from [47], we first propose a Dijkstra-like algorithm with an extended level of information; and (2) robust NF provisioning: we present an approximation algorithm for SFC-Fork and a mixed-integer programming formulation as its exact solution approach for the general cases.

2.2.1 | Facility location approximations

We briefly summarize some existing approximation algorithms for the k-level facility location problem and robust fault-tolerant facility location problem with a restriction on the standard uncapacitated facility location problem where no penalty is allowed. These related works would serve as the foundation for us to discuss the potential of the approximation algorithms for robust NF provisioning. The k-level facility location problem, which has a client set and k types/levels of facility sets, aims to connect clients to opened facilities at each level (in the order of level 1 to level k) with minimal costs/weights. Through LP relaxation, Guha and Khuller [27] generalize the 1-level uncapacitated facility location problem and develops an algorithm with an approximation ratio of 1.463. Krishnaswamy and Sviridenko [33] improve the ratio to 1.61 for general k. Harder than the 1-level facility location problem, the k-level facility location problem has a currently best-known approximation with a ratio of 1.488 [34]. The constant factor approximation is started from a 2-level problem ($k = 2$) with a 3.16-approximation algorithm [48], and a 3-approximation for general k [2]. Ageev et al. [4] demonstrate that the k-level problem can be reduced to a $(k-1)$ - level problem and a 1-level problem and provides a 2.43-approximation, which is further improved by [9, 11, 55] through LP primal-dual, randomization, and facility scaling techniques. Chechik and Peleg [13] introduce the robust fault-tolerance problem, which has a two-stage robust optimization setting with the first stage determining the facility location and client assignment, and the second stage reconnecting clients if up to a total of α connected facilities failed (which were opened in the first stage). Based on this setting, Chechik and Peleg [13] develop a $(7.5\alpha + 1.5)$-approximation algorithm, which is further improved to a $(k + 5 + k/4)$-approximation in [10] through LP-rounding.

3 | NOTATIONS AND PROBLEM DESCRIPTION

In this section, we first provide the general notations used in the discussion. We then propose the robust NF-service evaluation metric and the robust VNF provisioning problem to minimize the number of instantiated VNFs while maximizing the robust NF-service evaluation metric. Let $G_P = (V_P, A_P)/G_P = (V_P, E_P)$ be the physical infrastructure with node set V_P and arc/edge set A_P/E_P, required NF set F, and end-to-end service request $\mathcal{D} = \{d_{st}\}$. Let node set $V_P^f \in V_P$ denote a physical resource pool for NF f (candidate physical nodes to deploy f) and let $V_P^F = \cup_{f\in F} V_P^f$ be the NF-enabled node set. Each of the NF-enabled nodes has failure probability ρ_i, $i \in V_P^F$, and $0 \leq \rho_i \leq 1$.

We assume that the NF requests d_{st}, $s, t \in V_L$ are known a priori. Let d_{st} be a tuple $[(s, t), \sigma_{st}, F_{st}]$, where σ_{st} indicates whether the request is with SFC or not; if yes, $\sigma_{st} = 1$, otherwise $\sigma_{st} = 0$. We let $\widetilde{P}$ and $\overrightarrow{P}$ be the undirected and directed path sets in the physical network. A demand (s, t) with NF requests is *fulfilled* if it is routed through path $p_{st} \in \overrightarrow{P}$ visiting all required NFs in the sequence defined in SFC when $\sigma_{st} = 1$, or otherwise routed through undirected path $\eta_{st} \in \widetilde{P}$ visiting all required NFs with $\sigma_{st} = 0$. To simplify the notation, we let P represent the path set containing all undirected and directed paths of all NF requests.

TABLE 1 Notations and parameters

Notation	Description
$G_P(V_P, A_P)$, $G_P(V_P, E_P)$	Physical infrastructure G_P with V_P, A_P/E_P as its node and arc/link sets, respectively
$G_L(V_L, E_L)$	Logical network G_L with V_L, E_L as its node and link sets
P	The undirected and directed path set in G_P, where $\eta \in P$ and $p \in P$ denote the undirected and directed paths, respectively
F	The NF set, where $f \in F$ denotes a network function
V_P^F	A set of all NF-enabled nodes, $V_P^F \subseteq V_P$
$\Gamma(F)$	NF instance deployment, denoted as a tuple $[\{f, i, n_i^f\} : f \in F, i \in V_P]$, where n_i^f is the instances of f deployed onto i
Parameter	**Description**
$\mathcal{D}, d_{st}$	$\mathcal{D}$ is a set of service requests, where each $d_{st} \in \mathcal{D}$ is a tuple $[(s, t), \sigma_{st}, F_{st}]$ representing one service request; here F_{st} is the set of required NFs for demand (s, t), and $\sigma_{st} = 1$ if demand (s, t) is SFC (i.e., $f \in F_{st}$ should be executed in a fixed sequence), otherwise, $\sigma_{st} = 0$
ρ_i	The failure probability of NF-enabled physical node $i, i \in V_P^F$

Without loss of generality, we assume that the physical substrate network is at least 2-connected in this paper. Notations and parameters are summarized in Table 1.

3.1 | Robust NF service evaluation metric

Our robust NF-service evaluation metric is based on the following observations.

Observation 1. Given an NF-enabled node pool V_P^F and requests $\mathcal{D} = \{d_{st}\}$, where request d_{st} is realized through a path η_{st}. d_{st} cannot be fulfilled if and only if $V_P^f \cap \eta_{st} = \emptyset, f \in F_{st}$.

Observation 1 is derived from the fact that d_{st} can only be fulfilled if and only if (all) the required NFs are deployed onto physical node(s) in its selected path η_{st}.

If $F_{st} = \{f\}$ and η_{st} is given for all (s, t), $Prob(d_{st})$, the probability of d_{st} being fulfilled, is then $\left(1 - \prod_{i \in V_P^f \cap \eta_{st}} \rho_i\right)$. We now consider a more generalized setting where demands are with single or multiple NFs and their routings η_{st} are not given.

Definition 2. Given NF-enabled node pool V_P^F, the robust NF-service evaluation metric, denoted as $\mathcal{RP}(d_{st})$, is

$$\mathcal{RP}(d_{st}) = \min_{f \in F_{st}} \max_{\eta_{st} \in P_{st}} \left[1 - \prod_{i \in \Gamma(f) \cap \eta_{st}} \rho_i\right].$$

Note here that d_{st} with multiple non-chained NF requests is fulfilled if and only if all required NFs are satisfied. Thus, the robust evaluation metric $\mathcal{RP}(d_{st})$ is determined by the worst best-case scenario among all requested NFs realized through the best-known paths in P. Hence, $\mathcal{RP}$ provides an *estimated lower bound* on NF-service reliability for all demands.

Different from non-chained NF requests, SFC request is fulfilled only when all required NFs are served in a specified sequence. Without loss of generality, we assume that (1) the same NF request will not be fulfilled more than once on different NF-enabled nodes, and (2) each NF-enabled node will not carry out multiple NF requests in SFC.

Definition 3. Given NF-enabled node pool V_P^F, the robust NF evaluation metric of SFC request d_{st} is

$$\mathcal{RP}(d_{st}) = \min_{f \in F_{st}} \max_{p_{st} \in P_{st}} \left[1 - \prod_{i \in \Gamma(f) \cap p_{st}} \rho_i\right] / \mid F_{st} \mid !.$$

Since demands with SFC requests are fulfilled only when all requested NFs are deployed onto p_{st} and visited in a predefined sequence, there is only one valid case out of $|F_{st}|!$ permutations. $\mathcal{RP}(d_{st})$ is then determined by the worst best-case scenario among all requested NFs realized through the best-known paths in P (with the highest probability to survive).

Considering multiple NF requests in a given NFVI (NFV infrastructure) and managed by the same NFV MANO, we define the robust NF-service evaluation metric among all NF request as follows.

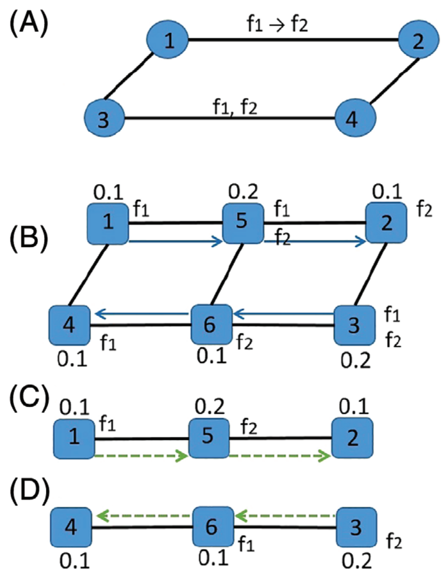

FIGURE 2 NF reliability [Color figure can be viewed at wileyonlinelibrary.com]

Definition 4. Given G_P, G_L, a set of NFs F, NF-enabled node pool V_P^F and node failure probability $\rho_i, i \in V_P^F$, $\mathcal{RP}(V_P^F) = \min_{d_{st} \in D} \mathcal{RP}(d_{st})$.

Naturally, as the counterpart of robust NF evaluation metric, we may derive the following properties.

$$\mathcal{FP}(d_{st}) = \max_{f \in F_{st}} \min_{\eta_{st} \in P_{st}} \left[\prod_{i \in \Gamma(f) \cap \eta_{st}} \rho_i \right], \tag{1}$$

$$\mathcal{FP}(d_{st}) = \max_{f \in F_{st}} \min_{\eta_{st} \in P_{st}} \left[\prod_{i \in \Gamma(f) \cap p_{st}} \rho_i \right] / \mid F_{st} \mid !, \tag{2}$$

$$\mathcal{FP}(V_P^F) = \max_{d_{st} \in D} \mathcal{FP}(d_{st}). \tag{3}$$

3.2 | Illustrations: NF service reliability versus robust NF service evaluation metric

We evaluate the robust NF-service evaluation metric via an instance illustrated in Figure 2 and present its differences from the NF-service reliability defined in [23]. In this example, two demands with NF requests d_{12} and d_{34} are considered. Demand d_{12} requires SFC $f_1 \rightarrow f_2$ and d_{34} requires non-chained NFs $\{f_1, f_2\}$. NF-enabled nodes, their supported NFs, and their failure probabilities are labeled in Figure 2. Candidate physical nodes to enable/deploy f_1's are in set $V_P^1 = \{1, 3, 4, 5\}$, and those for f_2's are in $V_P^2 = \{2, 3, 5, 6\}$. d_{12} is routed through a directed path $\{(1, 5), (5, 2)\}$, and d_{34} is routed through an undirected path $\{(4, 6), (6, 3)\}$. Based on the assumptions given in the previous section, the robust NF-service evaluation metric $\mathcal{RP}(\{d_{12}, d_{34}\}) = \min\{1 - 0.1, 1 - 0.2, (1 - 0.2 \times 0.1)/2, (1 - 0.1 \times 0.2)/2\} = 0.49$.

Different from $\mathcal{RP}(\{d_{12}, d_{34}\})$, NF-service reliability of d_{12} is $[1 - Prob(f_1, f_2$ both failed$) - Prob($only f_2 failed$) - Prob($only f_1 failed$) - Prob(f_1, f_2$ fulfilled but not in - order$)] = 1 - 0.1 * 0.2 * 0.1 - 0.9 * 0.2 * 0.1 - 0.9 * 0.2 * 0.1 - 0 = 0.962$. The NF-service reliability of $d_{34} = [1 - Prob(f_1, f_2$ both failed$) - Prob($only f_2 failed$) - Prob($only f_1 failed$)] = 1 - 0.2 * 0.1 * 0.1 - 0.2 * 0.1 * 0.9 - 0.2 * 0.9 * 0.1 = 0.962$.

The examples above show that NF-service reliability is measured when the deployment of NF instances and routings is determined. In contrast, since the robust NF-service evaluation metric already evaluates the minimum NF reliability, the routings selected and the deployment of non-chained NFs or SFC would always be better than or at least equal to the metric. In other words, the robust NF-service evaluation metric provides a tight lower bound for each NF's reliability.

This instance also shows that with the limitation imposed on the NF-enabled nodes, the selection of NF-enabled nodes also impacts the robust NF-service evaluation metric. Hence, in the following section, we study the robust NF provisioning problem which aims at maximizing our proposed NF-service evaluation metric via NF-enabled node selection for NF request realization.

3.3 | Robust NF routing

We study the subproblems in robust NF provisioning, the NF $s-t$ path, SFC $s-t$ path, and request routing problems with the assumption that all NF-enabled nodes are deployed with corresponding NF instances.

3.3.1 | NF path and SFC path problems

Different from typical $s-t$ routing problems in telecommunication networks, an SFC $s-t$ routing problem is to find a route realizing SFC request and guaranteeing that all required NFs are visited in order. The corresponding NF $s-t$ routing problem is to find a $s-t$ route visiting all required NFs enabled nodes. We formally define them as follows.

Definition 5. Given demand d, required NFs $F(d)$, physical substrate network $G_P(V_P, E_P)$, and NF enabled node set $N_P(f)$, a NF $s-t$ path problem is to find a $s-t$ path p_{st} where $N_P(f) \cap p_{st} \neq \emptyset$ with $f \in F(d)$.

Definition 6. Given demand d, its required SFC $(f_1, f_2, \ldots, f_r)$, physical substrate network $G_P(V_P, E_P)$, and NF enabled node set $N_P(f)$, an SFC $s-t$ path problem is to find a $s-t$ path p_{st} which satisfies $N_P(f) \cap p_{st} \neq \emptyset$ with $f \in F(d)$ and visits NF-enabled node in the same sequence given in SFC.

Applying the robust NF evaluation metric, correspondingly, we introduce the robust counterparts of the above two problems as follows. With a single source-destination pair, the robust NF and SFC $s-t$ paths aim to find a $s-t$ path maximizing the minimal successful rate among all required NFs, that is, $\max_{p\in P^S_{st}} \min_{i\in p} \ln[1+(1-\rho_i)]$ and $\max_{p\in P^F_{st}} \min_{i\in p} \ln[1+(1-\rho_i)]$, with P^S_{st} and P^F_{st} being the path sets for SFC $s-t$ path and NF $s-t$ path, respectively.

Given an SFC request, we consider its embedded SFC chain as a directed path in the logical network. While multiple demands with SFC requests are given, the required SFC chains together form an SFC forwarding graph (logical network). Let $G_L(V_L, E_L)$ indicate the logical SFC forwarding graph. In Section 4, we present a labeling-based pseudo-polynomial algorithm for the SFC $s-t$ path and NF $s-t$ path on the constructed auxiliary NF-enabled network which is an intermediate network layer in between the lower-layer physical substrate network (namely, the physical network) and the upper-layer SFC chain (namely, the logical network). Extending from a single source-destination pair of NF request, we consider the NF and SFC request routing problems in the next section.

3.3.2 | NF and SFC request routing

We still assume that all NF-enabled nodes are deployed with NF instances and study NF and SFC routing for all NF requests, whose corresponding general network design problem is the multicommodity flow problem.

Given NF requests $D = \{d\}$, physical substrate network $G_P(V_P, E_P)$, and NF enabled node set $N_P(f)$, the *NF request routing* and *SFC request routing* problems are to generate NF/SFC paths for all demands with NF/SFC requests, respectively. Through adopting the robust NF evaluation metric as the objective, the robust NF and SFC routing problems determine the routes for all NF requests while evaluating the NF failure rate among all NFs and requests.

3.4 | Robust NF provisioning

With the evaluation metric above, we present in the following the *VNF provisioning problem* without considering the NF-enabled node failures. Given G_P, G_L, $\mathcal{D}$, and V^F_P. $d_{st} \in \mathcal{D}, s, t \in V_L$, is mapped onto a directed path $p_{st} \in P$ for SFC request, or onto an undirected path $\eta_{st} \in P$ for NF request. We would like to determine a limited number of NF-enabled nodes to support each required NF and guarantee that demands are routed through their required NFs. This problem considers both non-chained NF and SFC requests.

When taking the failures of NF-enabled nodes into consideration, we now define the robust VNF provisioning problem.

Definition 7. Given N_f as the limited number of NF-enabled nodes supporting NF f, the *robust VNF provisioning problem* is to determine the NF deployment which maximizes the robust NF-service evaluation metric: $\max_{V^f_P:|V^f_P|\leq|N_f|} \mathcal{RP}(V^F_P)$.

3.5 | Maximizing $\mathcal{RP}(V^F_P)$ via minimizing $\mathcal{FP}(V^F_P)$

We show that the robust VNF provisioning can be achieved via finding the *minimum robust NF failure evaluation metric*.

TABLE 2 Parameters and variables

Parameter	Description
N_f	The number limitation of NF deployed locations with $f \in F$
ρ_i	The failure probability of physical node i with $i \in V_P$
$\delta^i_{\eta_{st}}$	A binary indicator showing whether physical node i is on path η_{st} or not, $\eta_{st} \in \mathcal{P}_{st}, (s,t) \in E_L$; if yes, $\delta^i_{\eta_{st}} = 1$, otherwise $\delta^i_{\eta_{st}} = 0$
γ^f_{st}	A binary indicator showing whether f is requested by d_{st} or not; if yes, $\gamma^f_{st} = 1$, otherwise, $\gamma^f_{st} = 0$
M	A very large number
Variable	**Description**
λ	The upper bound on NF failure probability for service requests in $\mathcal{D}$
ξ^f_{st}	NF failure probability of NF $f \in F$ and $d_{st} \in \mathcal{D}$
$x_{p_{st}}$	A binary variable indicating whether path $p_{st} \in \mathcal{P}_{st}$ is selected to fulfill $d_{st} \in \mathcal{D}$; if yes, $x_{p_{st}} = 1$, otherwise, $x_{p_{st}} = 0$
y^{if}_{st}	A binary variable indicating whether physical node i provides NF requests f for d_{st} or not; if yes, $y^{if}_{st} = 1$, otherwise, $y^{if}_{st} = 0$
h_i	A binary variable which indicates whether a network function is deployed onto physical node i or not; if yes, $h_i = 1$, otherwise, $h_i = 0$
z^f_i	A binary variable which indicates if network function f is deployed onto physical node i; if yes, $z^f_i = 1$, otherwise, $z^f_i = 0$
β_{st}	A binary auxiliary variable which indicates if demand d_{st} is selected under the SFC setting; if yes, $\beta_{st} = 1$, otherwise $\beta_{st} = 0$

Proposition 8. $1 - \mathcal{RP}(V^f_P) = \mathcal{FP}(V^f_P)$, *with* $d_{st} \in \mathcal{D}$ *and* $f \in F_{st}$.

Derived directly from Definitions 2 and 3, Proposition 8 also holds for SFC requests. Hence, we have the following conclusion.

Theorem 9. $\max_{V^F_P} \mathcal{RP}(V^F_P) = \min_{V^F_P} \mathcal{FP}(V^F_P)$.

In the next section, we demonstrate that solving the robust VNF provisioning via minimizing NF failure evaluation metric would linearize the nonlinear equations. We then propose the solution approach accordingly.

4 | SOLUTION APPROACH

In this section, we present solution approaches to solve the robust VNF provisioning problem. We start with two special cases/ subproblems: (1) the robust NF and SFC $s-t$ path problems, for which we construct an auxiliary network layer and present pseudo-polynomial algorithms for both; and (2) we leverage the k-level facility location problem and construct a 3.27-approximation algorithm for the VNF provisioning problem with SFC-Fork as the forwarding graph. We develop the two-step path-reduction technique and demonstrate the existence of the approximation algorithm for the robust VNF provisioning problem through SFC-Fork. For the general problem setting (not limited to the SFC-Fork structure), we demonstrate how to utilize $\mathcal{FP}(V^F_P)$ to formulate the robust VNF provisioning problem and propose its MILP solution approach. The variables and parameters used in this section are presented in Table 2.

4.1 | Special case 1: NF and SFC $s-t$ path problems

In this section, we present a pseudo-polynomial algorithm for the minimal weighted and robust non-chained NF $s-t$ path and SFC $s-t$ path problems, respectively. Vardhan et al. [49] proved that the problem of finding a path with multiple must-stop nodes and without order requirements is NP-Complete. The NF $s-t$ path is a path between s and t and must stop at NF enabled nodes; the SFC $s-t$ path problem further requires the NF path to visit NFs with defined order in SFC. Hence, we explore a pseudo-polynomial algorithm for the NF and SFC $s-t$ path problems.

4.1.1 | Auxiliary NF-enabled network construction

We first introduce a condensed physical network, an *auxiliary NF-enabled network*, which only contains source and destination nodes of NF requests and their corresponding NF-enabled nodes. To allow a single node in the physical substrate network to support multiple types of NFs, we introduce augmentation steps that create copies of NF-enabled nodes and indicate their supported types of NFs in Algorithm 1.

Algorithm 1. Node set construction in the auxiliary NF-enabled network

---

Input: Physical substrate network $G_P(V_P, E_P)$, NF-enabled node set $N_P(f)$ with $f \in F$, and the initial augmented NF-enabled node set $V_P^A = \emptyset$
Output: An augmented NF-enabled node set

```
for i ∈ N_P and f ∈ F do
  if i ∈ N_P(f) then
    Create a copy of i indicated as i_f
    V_P^A = V_P^A ∪ i_f
  end if
end for
```

We next present an algorithm that adds arcs in the auxiliary NF-enabled network through the cross-layer network concept, where we consider the SFC chain or SFC forwarding graph as the upper-layer/logical network and the physical substrate network as the lower-layer/physical network. After introducing duplicated NF-enabled nodes and their available NFs support, we build connections among these NF-enabled nodes based on the service requests. To limit the size of the augmented network, we only add arcs connecting nodes in V_P^A when the connection can realize the NF or SFC routes. We wish to note that the connectivity of the auxiliary NF-enabled network for non-chained NF requests is higher than that of the SFC version as the non-chained NF requests do not require in-order execution.

Algorithm 2. Arc construction with SFC requests in the auxiliary NF-enabled network

---

Input: Physical substrate network $G_P(V_P, E_P)$, NF-enabled node set $N_P(f)$ with $f \in F$, SFC forwarding graph $G_L(V_L, E_L)$, augmented NF-enabled node set V_P^A and the initial augmented NF-enabled arc set $E_P^A = \emptyset$
Output: Arc set for auxiliary NF-enabled network E_P^A for SFC requests

```
for e = (f_i, f_j) ∈ E_L do
  for ℓ ∈ N_P(f_i) and k ∈ N_P(f_j) do
    if a path ρ(ℓ, k) exists in G_P then
      Create arc (ℓ(f_i), k(f_j)) and add the arc into E_P^A
    end if
  end for
end for
```

To differentiate the auxiliary NF-enabled network for NF requests and SFC requests, we let $G_S(N_S, E_S)$ and $G_F(N_F, E_F)$ denote networks for non-chained NF requests and SFC requests, respectively, where $N_S = V_P^A, N_F = V_P^A$, and E_S and E_F are obtained through Algorithm 2 and 3, respectively. We illustrate an instance of the auxiliary NF-enabled network for SFC in Figure 3, which is an abstraction of all potential SFC path realization via NF-enabled physical nodes. Given SFC (f_1,f_2,f_3) (see Figure 3A), where f_1, f_2, and f_3 are with 2, 3, and 2 NF-enabled physical nodes, respectively, which are illustrated in Figure 3B. All possible SFC physical paths for (f_1,f_2,f_3) through their corresponding NF-enabled nodes can be calibrated. For instance, there are 12 possible physical paths to realize the SFC in Figure 3.

Proposition 10. *Given a constructed $G_S(N_S, E_S)$, the visiting sequence of NFs in SFCs is realized through the arcs in E_S.*

Proof. We prove this claim by contradiction. Given an SFC λ and its corresponding $G_S(N_S, E_S)$ and let (f_i, f_j) and (f_j, f_k) be arcs in λ. We assume that E_S, arc (i_{f_i}, i_{f_k}) exists. Based on Algorithm 2, only (i_{f_i}, i_{f_j}) and (i_{f_j}, i_{f_k}) are created. Hence, connecting i_{f_i}, i_{f_k} requires at least two arcs in G_S. Contradiction! ■

Algorithm 3. Arc construction with non-chained NF requests in the auxiliary NF-enabled network

Input: Given physical substrate network $G_P(V_P, E_P)$, NF-enabled node set $N_P(f)$ with $f \in F$, demand d_{st} with required NFs $D(F)$, augmented NF-enabled node set V_P^A, and the initial augmented NF-enabled arc set $E_P^A = \emptyset$
Output: Arc set for auxiliary NF-enabled network E_P^A for non-chained NF requests

for any two $f_i, f_j \in D(F)$ **do**
  for $\ell \in N_P(f_i)$ and $k \in N_P(f_j)$ **do**
    if a path $\rho(\ell, k)$ exists in G_P **then**
      Create arc $(\ell(f_i), k(f_j))$ and add the arc into E_P^A
    end if
  end for
end for
$G_S = G_S \cup \{s, t\}$
$\rho_s = 0$, $\rho_t = 0$
for $i \in N_P(f_1)$ **do**
  if a path $\rho(s, i)$ exists in G_P **then**
    Create arcs (s, i) and add into E_S
  end if
end for
for $i \in N_P(f_r)$ **do**
  if a path $\rho(i, t)$ exists in G_P **then**
    Create arcs (i, t) and add into E_S
  end if
end for

4.1.2 | Pseudo-polynomial algorithm for SFC $s-t$ path

We first define the SFC $s-t$ path problem before presenting the algorithm. Given physical substrate network $G_P(V_P, E_P)$ and all of its supported NFs, we let $N_P(f)$ represent an NF-enabled physical node set supporting NF $f \in F$.

Definition 11. Given an SFC chain, an *SFC service path* is a physical path connecting NF-enabled nodes with required NFs following the sequence defined in the SFC chain.

We let $\mathcal{P}(f_h)$ represent a subpath set of SFC physical paths in $G_P(V_P, E_P)$, which starts from physical nodes in $N(f_h)$ and ends at physical nodes $N(f_r)$.

We now present the algorithm for SFC $s-t$ path. Given G_P, G_S, G_L and the source and destination nodes s and t of the SFC. For the general minimal-weighted SFC $s-t$ path problem, we first add s and t connecting the first and last NF-enabled nodes in G_S. Here, the weight can be the shortest-path weight in the physical substrate network. Different from the minimal-weighted $s-t$ path, the SFC $s-t$ path should visit the required NFs in the order specified in SFC. Since Proposition 10 shows that the order of NFs in SFC is preserved in the auxiliary NF-enabled network, we present a Dijkstra-like algorithm for SFC $s-t$ path as follows.

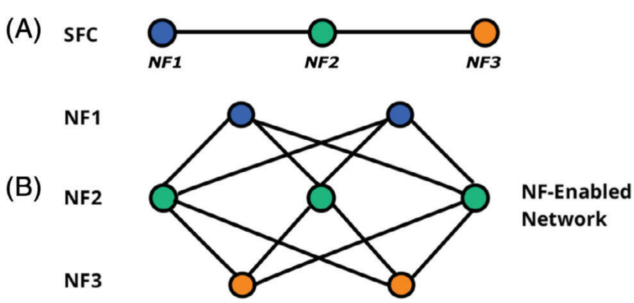

FIGURE 3 Auxiliary NF-enabled network for SFC [Color figure can be viewed at wileyonlinelibrary.com]

Algorithm 4. SFC $s-t$ path algorithm

Input: Given $G_P(V_P, E_P)$, $G_S(N_S, E_S)$, and SFC chain $G_L(V_L, E_L)$, source and destination node s, t
Output: SFC path ρ
for $i \in N_S$ **do**
  Set initial visited ancestor list $\ell(i) = \emptyset$
end for
Set a node set $U = \{s\}$, $\omega(s) = 0$
while $U \neq V_P$ **do**
  for j in U's adjacent nodes **do**
    $\omega(j) = \min_{e=(v,j):v \in U}[\omega(v) + \omega(e)]$
  end for
  Add $j^* = \mathrm{argmin}_{j \in U'\text{s adjacent nodes}} \omega(j)$ to U
end while

4.1.3 | Pseudo-polynomial algorithm for robust SFC $s-t$ path

We present in this section the pseudo-polynomial algorithm for the robust SFC $s-t$ path problem. We identify the property of the robust SFC $s-t$ path and its non-chained counterpart as follows.

Proposition 12. *Given an auxiliary NF-enabled network $G_S(N_S, E_S)$ and $G_F(N_F, E_F)$*

1. *The robust SFC $s-t$ path problem is an SFC $s-t$ bottleneck path problem in $G_S(N_S, E_S)$.*
2. *The robust non-chained NF $s-t$ path problem is a non-chained NF $s-t$ bottleneck path problem in $G_F(N_F, E_F)$.*

Proof. The $s-t$ bottleneck path problem determines a path with a maximal path capacity defined as the minimal edge capacity on the path. Let all arcs in G_P have failure probability 0. Then, we do the typical arc augmentation for all NF-enabled nodes, where augmented nodes are added and directed arc are created to connect these nodes. For the SFC path, the arc direction follows the SFC chain; as for the non-chained NF path, the arcs are bi-directly generated. All augmented arcs have capacity $ln[1 + (1 - \rho_i)]$. Hence, the robust SFC $s-t$ and non-chained NF $s-t$ paths become the corresponding bottleneck path problems. ∎

Algorithm 5. Robust SFC $s-t$ path algorithm

Input: Given $G_P(V_P, E_P)$, $G_S(N_S, E_S)$, and SFC chain $G_L(V_L, E_L)$, source and destination node s, t
Output: SFC path ρ
for $e \in E_S$ **do**
  Set $\rho_e = -1$
end for
for all NF-enabled nodes in N_S **do**
  Augment all NF-enabled nodes as arcs and add into E_S
  Set augmented arc capacity $\omega(e) = ln(1 + (1 - \rho_i))$
end for
Set a node set $U = \{s\}$ and $\omega(s) = 0$
while $U \neq V_P$ **do**
  for j in U's adjacent nodes **do**
    $\omega(j) = \max_{e=(v \in U, j}[\min(\omega(v), \omega(e))]$
  end for
  $U = U \cup \{i\}$, while $i = \arg\min_{j \in U'\text{s adjacent nodes}} \omega(j)$
end while

For non-chained NF requests, we assume that only nodes in the auxiliary NF-enabled network have nodal weights and all arcs have weight equal to 0. We show that the optimal solution of an SFC $s-t$ path is also the optimal solution of its non-chained counterpart using the objective $\min\sum_{i\in\cup_{f\in F}i_f}c_ix_i$ which minimizes the node-weighted SFC $s-t$ path. We demonstrate that if NF-enabled nodes are reached to achieve the optimal solution, the visiting order of these nodes would not impact that optimal solution.

Proposition 13. *Given $G_S(N_S,E_S)$ and two $s-t$ NF paths p_1 and p_2 containing NF-enabled nodes N_1 and N_2, where $p_1\neq p_2$ if $N_1=N_2$. We have* $\min\sum_{i\in\cup_{f\in F}i_f\cap p_1}c_ix_i=\min\sum_{i\in\cup_{f\in F}i_f\cap p_2}c_ix_i$.

Hence, we define an SFC for the non-chained NF $s-t$ path and apply Algorithm 4 and 5 to obtain the (robust) non-chained NF $s-t$ path.

4.2 | Special case 2: robust NF provisioning with SFC-Fork

In this section, we present an approximation algorithm for robust VNF provisioning with *SFC-Fork* (also denoted as *SFork*). Note here that SFC-Fork is a common NF forwarding graph defined in practice, which has a rooted tree structure with a single branching point. We first review the existing k-level facility location bi-factor approximation algorithm and demonstrate that VNF provisioning with a single SFC can be reduced to a k-level facility location problem. We then design an approximation algorithm for the problem. To manage the SFC-Fork as the NF forwarding graph, we apply a two-step parameterized path reduction in the bi-factor approximation algorithm and prove that it is a 3.27-approximation. Leveraging the approximation algorithm for robust facility location problems, we further demonstrate the existence of the approximation algorithm for the robust VNF provisioning problem.

We let $G_S(V_S,A_S)$ represent an SFC forwarding graph, $(f_1,f_2,\ldots,f_r)$ be an SFC chain, and $\Lambda=\{\lambda\}$ denote a set of SFC chains with $\lambda\in\Lambda$ as an SFC chain. Given physical substrate network $G_P(V_P,E_P)$ and all of its supported NFs, we let $N_P(f)$ represent an NF-enabled physical node set supporting NF $f\in F$. We let $\mathcal{P}(f_h)$ represent a subpath set of SFC physical paths in $G_P(V_P,E_P)$, which starts from physical nodes in $N_P(f_h)$ and ends at physical nodes $N_P(f_r)$.

Assumption 14. We assume that $G_P(V_P,E_P)$ is at least two-connected.

With Assumption 14, the networks created through Algorithm 1–3 are at least two-connected.

Proposition 15. *With Assumption* 14, *if* $|N_P(f)|\geq 2$ *for* $f\in F$, $G_S(V_S,E_S)$ *is at least two-connected.*

4.2.1 | Review on k-level facility location problem

Given a client set D and a facility set F_ℓ at level ℓ, the k-level facility location problem determines the sets of facilities $X_\ell\in F_\ell$ to be opened at level $1\leq\ell\leq k$ and connects client $d\in D$ to a *facility service path* $(i_k(d),i_{k-1}(d),\ldots,i_1(d))$ with facility location at $i_k(d)$. The 1-level facility location problem only has a single level of facility set, where all clients are directly connected to the facility location without a service path.

Ageev et al. [4] demonstrate that an instance of k-level facility location problem can be reduced to an instance of 1-level facility location problem as follows: the 1-level problem takes the client set in the k-level problem as its client set, and the facility location set is determined by the potential facility service paths from level k to level 1, which is denoted as

$$\rho(i_k,t)=\arg\min_{\rho\in\mathcal{P}_F}\{t\times\beta\times C(\rho)+\alpha\times O(\rho)\},$$

where $\mathcal{P}_F$ is an NF service path set and $t=1,\ldots,|D|$, $i_k\in X_k$. Client $j\in D$ can be connected to these determined service paths with connection cost $C(j,i_k)+C(\rho(i_k,t))$. Given a solution of the 1-level facility location problem (denoted as SOLS) constructed above, a corresponding k-level facility location solution (denoted as SOLM) can be constructed through opening all facilities on above facility services paths and connecting all clients with their corresponding service paths.

Theorem 16. (Theorem 1 in [4]). *If SOLS is an (α,β)-approximate solution of an 1-level facility location instance, then, SOLM is a $(\alpha,3\beta)$-approximate solution of a k-level facility location instance.*

Next, we present approximation algorithms for NFP-SFork starting with a simple case, where a single SFC is the NF-forwarding graph. In other words, all SFC requests require the same SFC. Extending from this special case, we present approximation algorithms with SFork as the forwarding graph in both deterministic and robust settings.

4.2.2 | Approximation algorithm for NFP-1SFC

We first study a special case of the NF provisioning problem, where all requests require the same SFC $\lambda = (f_1, f_2, \ldots, f_\gamma)$ and γ indicates the γth NF in the SFC. We denote the problem as NFP-1SFC.

NFP-1SFC can be reduced to a k-level facility location problem through the following steps. First, we convert an SFC path set into the connection between a request and its SFC service path, where required NFs specified in the SFC should be visited in order along the path. We calibrate the cost of request $d \in D$ to its SFC service path with a simple reduction. The connection cost for a request $d(s, t)$ to a path SFC service ρ starting from i_{f_γ} and ending at i_{f_1} is

$$C((s, i_{f_1}), \rho, (i_{f_\gamma}, t)) = C(s, i_{f_1}) + C(\rho) + C(i_{f_\gamma}, t) = C(d, i_{f_1}, i_{f_\gamma}, \rho) + C(\rho),$$

with $C(d, i_{f_1}, i_{f_\gamma}, \rho) = C(s, i_{f_1}) + C(i_{f_\gamma}, t), i_{f_1}, i_{f_\gamma} \in \rho$ and $\rho \in \mathcal{P}_F$. The procedure is presented in Algorithm 6. Based on $G_S(V_S, E_S)$'s 2-connectivity given in Proposition 15, the connection between the source and destination nodes of a request and a service path' two-end nodes can be established.

Algorithm 6. SFC routing conversion to request and service path connections

Input:
for $d = (s, t) \in D, i_{f_1} \in X_1$, and $i_\gamma \in X_\gamma$ **do**
  for $\rho \in \mathcal{P}_F$ **do**
    Calibrate the shortest path between (s, i_{f_1}) and (t, i_{f_r}) in G_P with $i_{f_1}, i_{f_r} \in \rho$
    Set $C(d, i_{f_1}, i_{f_r}, \rho) = C(s, i_{f_1}) + C(i_{f_r}, t)$ end for
end for

After applying Algorithm 6, the NFP-1SFC reduces to a problem that (1) determines NF-enabled nodes at each level for NF instance deployment, and (2) guarantees all requests are connected to an NF service path visiting NFs in the order defined in the SFC. Hence, the problem is a γ-level facility location problem, where (1) the NF request set is the client set, and (2) the NF-enabled node set corresponding to NF i is the level i facility set ($1 \le i \le \gamma$). The difference between the two is that the connection cost of a request to a service path is composed of two parts, namely $C(s, i_{f_1}) + C(i_{f_r}, t)$. Different from the k-level facility location problem, we let $\overline{\mathcal{P}}_S$ be the service path set for request $d(s, t)$ which starts and ends at node i_{f_r} and i_{f_1}, respectively, with $i_{f_r} \in X_\gamma$ and $i_{f_1} \in X_1$. We select service path $\rho(t, i_{f_1}, i_{f_\gamma}, d)$ for request $d \in D$ as

$$\arg\min_{\rho \in \overline{\mathcal{P}}_S} \{t \times \beta \times C(\rho) + \alpha \times O(\rho)\},$$

where $t = 1, \ldots, |D|$. We let request $d(s, t)$ connect to NF service path $\rho(t, i_{f_1}, i_{f_\gamma}, d)$.

We further reduce NFP-1SFC to the 1-level facility location problem, where the SFC request set is taken as the client set, and the selected SFC service path set represents the facility set. Given a feasible solution of the 1-level facility location problem, denoted as ψ_{1FL}, we construct a solution for NFP-1SFC problem as follows—all NF-enabled nodes on selected SFC service paths are deployed with NF instances, and the demand $d(s, t)$ is connected to a selected service path.

With Theorem 16, the following conclusions hold.

Lemma 17. *Given an SFC chain $\lambda = \{f_1, f_2, \ldots, f_\gamma\}$ and a feasible solution ψ_{1FL},*

1. *if there exist NF service paths $\rho_1(\psi_{1FL}) = (i_{f_1}, i_{f_2}, \ldots, i_{f_\gamma})$, $\rho_2(\psi_{1FL}) = (i'_{f_1}, i'_{f_2}, \ldots, i'_{f_\gamma})$ and $i_{f_j} = i'_{f_j}$, another solution ϕ also exists based on SOLS, where NF service paths are*

$$\rho_1(\phi) = (i_{f_1}, i_{f_2}, \ldots, i_{f_j}, \ldots, i_{f_\gamma}) \text{ and } \rho_2(\phi) = (i'_{f_1}, i'_{f_2}, \ldots, i'_{f_j}, \ldots, i'_{f_\gamma}), \text{ with } i_{f_\ell} = i'_{f_\ell}, \ \ell \le j;$$

2. $C_1^{\mathrm{SOLS}} = C_1^{\phi}$; $O^{\mathrm{SOLS}} = O^{\phi}$; *and* $\sum_{i=2}^{\gamma} C_i^{\mathrm{SOLS}} \le \sum_{i=2}^{\gamma} C_i^{\phi}$.

We derive from the first claim that the solutions of the NFP-1SFC problem have SFC service paths satisfying all NF requests $(f_1, f_2, \ldots, f_r)$, which form a forest beginning at the first NF-enabled node. Hence, a feasible solution for NFP-1SFC also has the forest structure. Meanwhile, all NF-enabled nodes and arcs in the forest, formed by its SFC service paths, are unique. The NF deployment cost for NF provisioning is the total cost to deploy all NF-enabled nodes in the forest, and the connection costs are the total costs of arcs in the forest. Moreover, with SFC path selection, the Claim 2 further identifies that demands connected to a rooted tree of the forest have their source nodes connected to the root node. Hence, we have a similar conclusion on the bi-factor approximation for NF provisioning with a single SFC.

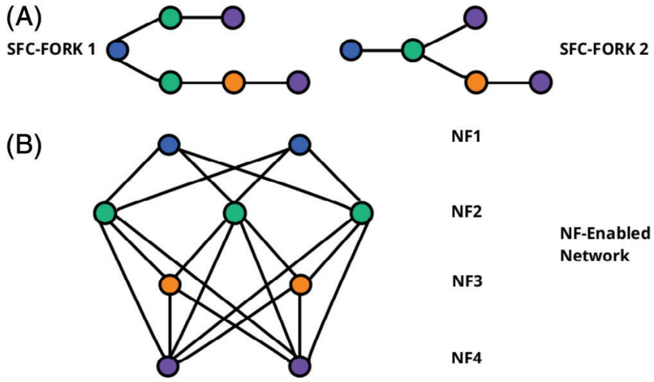

FIGURE 4 Fork SFC forwarding graph and NF-enabled node connection graph [Color figure can be viewed at wileyonlinelibrary.com]

Theorem 18. *If the* 1*-level facility location problem has a* (α, β)*-approximation solution, a solution of NFP-*1*SFC can be constructed through a* $(\alpha, 3\beta)$*-approximation.*

The detailed proof of Theorem 18 is very similar to that of Theorem 16 in [4]. Since NFP-1SFC is a special case of NFP-SFork, we include the proof of Theorem 20 in Appendix for NFP-SFork.

4.2.3 | Approximation algorithm for NFP-SFork

With multiple SFCs, the corresponding SFC forwarding graph identified in [15, 32, 38, 42] forms a fork network structure. Extending from NFP-1SFC, we study the approximation algorithm for SFC-Fork, where multiple SFC chains share the common structure. We define the SFC-Fork as follows.

Definition 19. Given $d \in D$ and all requested SFCs in Λ, an SFC-Fork is a rooted tree with a single branching node f_b, where $f_b \in \cup_{\lambda \in \Lambda} \cup_{f \in \lambda} f$.

Figure 4 illustrates two instances of SFC-Forks, where (1) the (sub)paths in SFC-FORK1 share a common NF—the root node of the fork; and (2) the SFC-FORK2 is branched at the NF2 (green) node. All SFCs visit the same NF with the same order. For example, NF 2 is visited before NF 4 for all SFCs. In other words, edges going from NF 4 to NF 2 are not allowed (only top-down order as in Figure 4B). Hence, a forwarding graph of an SFC-Fork is $\cup_{\lambda \in \Lambda} \lambda$. The difference between NFP-SFork and the k-level facility location problem is that the latter only contains a type of facility service paths, while NFP-SFork requires multiple types of SFC paths with shared NFs. Thus, NFP-SFork requires separate management to avoid duplicated counts either in its solution or its reduced 1-level facility location problem on NF deployed nodes. We now show that a bi-factor $(\alpha, 3\beta)$-approximation algorithm exists for NFP-SFork if the 1-level facility problem has a (α, β)-approximation. $(\alpha, 3\beta)$-approximation algorithm.

Given an instance of NFP-SFork $\mathcal{M}$, we let the common subpath of SFCs be $\rho_\Lambda^S = (f_1, \dots, f_b)$. If $b = 1$, only the first NF is shared; otherwise a subpath $(f_1, \dots, f_b)$, $2 \le b \le \min_{\lambda \in \Lambda} \gamma(\lambda)$, is shared, where $\gamma(\lambda)$ indicates the total number of NFs in SFC $\lambda \in \Lambda$. We reduce the NFP-SFork to the 1-level facility location problem through the following approaches: (1) facility set: reducing the shared sub-SFC service paths, and (2) client set: aggregating SFC requests and their remaining subpaths in SFC paths.

Algorithm 7 is a two-step parameterized path reduction algorithm for creating SFC service paths.

Algorithm 7. Two-step parameterized path reduction algorithm

Step 1: *Parameterized path reduction of SFC service subpaths from* $f_{b+1}(\lambda)$ *to* $\gamma(\lambda)$ *with* $\lambda \in \Lambda$. Let $\overline{\mathcal{P}}(b+1, \gamma(\lambda))$ be a path set on the auxiliary NF-enabled node network, G_A, connecting f_{h+1}-enabled nodes with $b+1 \le h \le \gamma(\lambda)$ and

$$p(t, i_{f_{b+1}}) = \arg \min_{p \in \overline{\mathcal{P}}(b+1, \gamma(\lambda))} \{t\beta[O(p) + C(p)]\}, t = 1, \dots, |D|.$$

This path set is called the **disjoint SFC service subpath**, where $O(p)$ and $C(p)$ are NF deployment costs and connection costs for path p, respectively.

Step 2: *Parameterized path reduction of the joint path among all SFCs*. We combine the joint path and subpaths in $\overline{\mathcal{P}}(b+1, \gamma(\lambda))$ and construct full SFC paths for requests. We determine the shared path as $p(j, i_{f_1}) = \arg\min_{p \in \overline{\mathcal{P}}(1,b)} \{\alpha O(p) + j\beta C(p)\}$ with $j = 1, \dots, |N(f_{b+1})|$, where $\overline{\mathcal{P}}(1, b)$ is the subpath set connecting the first NF-enabled nodes all the way through the f_b-enabled nodes in G_A. Hence, given a request $d \in D$, a feasible solution of NFP-SFork has the following structure – request d connects to path $p(t, i(f_{b+1}))$ and $p(j, i_{f_1})$, where t is the request, d is an index, and j is the index of the f_{b+1}-enabled node (denoted as $i(f_{b+1})$).

The shared subpath among requests is called the shared SFC service subpath for all $j \in N(f_b)$. The procedure of NFP-SFork reduction to 1-level facility location is now presented as follows.

1. Facility location set: it contains all $p(j, i_{f_1}), j = 1, \dots, \overline{P}(1, b)$, and has setup costs equal the sum of deployment and connection costs of their corresponding paths.
2. Client set: NF $(b+1)$-enabled nodes, $i_{f_{b+1}}$, where subpaths and SFC requests connecting to them are aggregated with $i_{f_{b+1}} \in X_{b+1}$. The connection cost to a facility location is the sum of (1) the connection costs from request d to a disjoint subpath, (2) the connection costs from disjoint SFC subpath to shared subpath, and (3) the deployment and connection cost of disjoint SFC service subpath, which is captured through

$$C(d, t) + C(p(t, i(f_{b+1})) + C(p(i_{f_{b+1}}, i_{f_1})) + C(i_{f_{b+1}}, i_{f_1}) + F(p(t, i(f_{b+1})).$$

We let O^1 indicate the total NF deployment cost corresponding to 1-level facility location facility cost, and O^2 indicates the total NF deployment cost which is part of 1-level facility location connection costs. Based on an instance of the 1-level facility location problem constructed above and one of its feasible solutions, we get an NFP-SFork feasible solution by

1. Following the facility and client connections—connect (1) the shared SFC service subpaths and disjoint SFC service subpaths, and (2) SFC requests to its disjoint subpaths.
2. Deploying NFs onto NF-enabled nodes for both shared and disjoint SFC service subpaths.

We now discuss the existence of a $(\alpha, 3\beta)$-approximation algorithm for NFP-SFork. The proof of Theorem 20 is given in the Appendix.

Theorem 20. *Given a (α, β)-approximation solution for the 1-level facility location problem, there exists a $(\alpha, 3\beta)$-approximation algorithm for the NFP-SFork problem.*

If the following inequalities hold individually, it sequentially leads to Theorem 20, where $\varphi_{1\text{fl}}$ is a feasible solution of the 1-level facility location problem, and φ_{sfc} is a feasible solution constructed based on $\varphi_{1\text{fl}}$ for NFP-SFork.

$$O(\varphi_{\text{sfc}}) + C(\varphi_{\text{sfc}}) \leq F(\varphi_{1\text{fl}}) + C(\varphi_{1\text{fl}}) \tag{4}$$

$$\leq \alpha F(\psi_{1\text{fl}}) + \beta C(\psi_{1\text{fl}}) \tag{5}$$

$$\leq \alpha O(\psi_{\text{sfc}}) + 3\beta C(\psi_{\text{sfc}}). \tag{6}$$

Based on the assumption that the 1-level facility location problem has a (α, β)-approximation algorithm, inequality (5) holds. The proofs of inequalities (4) and (6) are given in Lemmas 23 and 25, respectively in Appendix, where a supporting conclusion presented in Lemma 24 of Appendix shows that a forest structure exists in the joint SFC paths for requests of an SFC $\lambda \in \Lambda$.

Next, we present the bi-factor algorithm for NFP-SFork as follows.

Algorithm 8. Bi-factor approximation algorithm for NFP-SFork

Greedy Algorithm [4]

Step 1: Given a single level facility location problem, we scale the facility open cost up with a ratio δ with $\delta \geq 1$

Step 1.1: Initially, set $B_j = 0$ for all clients. Assign budget B_j to all clients j, and client j offers $\max\{B_j - c_{ij}, 0\}$ to facility i if j is not connected, otherwise, $\max_{i' \neq i}\{c_{i'j} - c_{ij}, 0\}$ if the client j connects to a facility i'.

Step 1.2: If unconnected client set $I_U \neq \emptyset$, increase B_j at the same rate; if the total offered costs to unopened facility is equal to open costs, i.e., $\sum_{j \in I_U} \max\{B_j - c_{ij}, 0\} = f_i$, open facility i; and if the connection costs of unconnected client j equals its connection cost to an opened facility i' $\max_{i' \neq i}\{c_{i'j} - c_{ij}, 0\} = c_{i'j}$, connect client j to facility i.

Step 2: Scale down the open costs of facilities to their original costs at the same rate; if opening a facility does not increase the total cost, the facility is open and assign clients to its closest open facility.

Algorithm 7: two-step path reduction

A greedy algorithm presented in [4] is a bi-factor $\gamma_f(\delta)$, $\gamma_c(\delta)$ approximation algorithm for the single-level facility location problem, where $\gamma_f(\delta) = \gamma_f + \ln(\delta)$ and $\gamma_c(\delta) = 1 + \frac{\gamma_c - 1}{\delta}$ with $\gamma_f = 1.11$ and $\gamma_c = 1.78$. Combining the greedy algorithm in [4] and the proposed Algorithm 7, we obtain a bi-factor approximation algorithm (8) for NFP-SFork.

Theorem 21. Algorithm 8 *is a* 3.27 *approximation algorithm for NFP-SFork problem regardless of the forward graph's network structure.*

Based on Theorem 20 and the $(\gamma_f(\delta), \gamma_c(\delta))$-approximation algorithm for 1FL, we have a $(\gamma_f(\delta), 3\gamma_c(\delta))$-approximation algorithm for NFP-SFork problem, regardless of the forwarding graph's fork structure with any $\delta \geq 1$. When $\delta = 8.67$, we obtain a feasible solution for NFP-SFork, which is within a factor of 3.27 of the optimal solution of NFP-SFork.

4.2.4 | Extension: robust NFP-SFork

To guarantee that there exists backup NF-enabled nodes after any NF-enabled node failure, we engage the robust fault-tolerance algorithm to ensure the availability of backup NF-enabled nodes, where the backup costs is also minimized at the level ℓ with $1 \leq \ell \leq \gamma(\lambda)$ with $\lambda \in \Lambda$.

Chechik and Peleg [14] concluded the existence of $(1.5 + 7.5\alpha)$-approximation for the robust fault-tolerant facility location problem with α failed nodes. Hence, applying two-step parameterized path reduction to create SFC service path and backup paths after NF-enabled node failure into the approximation algorithm for robust fault-tolerant facility location problem, an approximation algorithm exists for robust NFP problem with α NF-enabled node failure.

4.3 | Formulations for robust NF provisioning with NF request

We now present the mathematical formulations for the maximal reliable NF deployment problem based on the NF service failure probability. We first turn the nonlinear objective $\min_{V_P^F} \max_{d_{st} \in \mathcal{D}} \min_{\eta_{st} \in \mathcal{P}_{st}} \Pi_{i \in V_P^f \cap \eta_{st}} \rho_i$ into its linearized counterpart

$$\min_{V_P^F} \max_{\substack{f \in F_{st} \\ d_{st} \in \mathcal{D}}} \min_{\eta_{st} \in \mathcal{P}_{st}} \sum_{i \in V_P^f \cap \eta_{st}} \ln(1 + \rho_i) \tag{7}$$

by applying the $\ln(\cdot)$ function.

With Theorem 9, the formulation presented below is the robust NF evaluation metric value of NF request with (7) as the objective:

$$\min_{\lambda, x, y, z, \xi, h} \lambda$$

$$\begin{aligned}
\text{s.t.} \sum_{i \in V_P} h_i &\leq N_f, && f \in F, && (8)\\
\lambda &\geq \xi_{st}^f, f \in F, && d_{st} \in \mathcal{D}, && (9)\\
\xi_{st}^f &= \sum_{i \in V_P} \ln(1 + \rho_i) y_{st}^{if}, && f \in F,\ d_{st} \in \mathcal{D}, && (10)\\
y_{st}^{if} &\geq z_i^f + \delta_{\eta_{st}}^i x_{\eta_{st}} + \gamma_{st}^f - 2, && f \in F,\ d_{st} \in \mathcal{D},\ \eta_{st} \in \mathcal{P}_{st},\ i \in V_P, && (11)\\
y_{st}^{if} &\leq z_i^f, && f \in F,\ i \in V_P, && (12)\\
y_{st}^{if} &\leq \delta_{\eta_{st}}^i x_{\eta_{st}}, && f \in F,\ i \in V_P,\ d_{st} \in \mathcal{D},\ \eta_{st} \in \mathcal{P}_{st}, && (13)\\
y_{st}^{if} &\leq \gamma_{st}^f, && f \in F,\ i \in V_P,\ d_{st} \in \mathcal{D}, && (14)\\
h_i &\geq z_i^f, && f \in F,\ i \in V_P, && (15)\\
\sum_{\eta_{st} \in \mathcal{P}_{st}} x_{\eta_{st}} &= 1, && d_{st} \in \mathcal{D}, && (16)\\
\lambda, \xi_{st}^f \geq 0, z_i^f,\ & y_{st}^{if}, h_i,\ x_{\eta_{st}} \in \{0,1\},\ \eta_{st} \in \mathcal{P}_{st}, && (s,t) \in E_L, f \in F, d_{st} \in \mathcal{D}, i \in V_P && (17)
\end{aligned}$$

Constraint (8) enforces the upper bound for the number of nodes deployed with NFs. Constraint (9) records the value of NF failure evaluation metric (linearized) among all demands for all NFs. Constraint (10) captures the robust NF failure evaluation metric value (linearized, i.e., $\ln(1 + \rho_i)$ as in constraint (7)) of demand $d_{st} \in \mathcal{D}$ and $f \in F$. Based on Definition 2, constraint (11) determines whether f is deployed onto physical node i for demand $d_{st} \in \mathcal{D}$, where (i) $z_i^f = 1$ when f is deployed onto physical node i; (ii) $\delta_{\eta_{st}}^i = 1$ when node i deployed with an NF is on a selected path η_{st} for d_{st}; and (iii) $\gamma_{st}^f = 1$ when d_{st} requires NF f. Constraints (12)-(14) force variable y_{st}^{if} to be 0 when any of the (i) to (iii) above is not satisfied. Constraint (15) indicates whether physical node i is deployed with any NFs. Constraint (16) selects a single physical route for demand $d_{st} \in \mathcal{D}$. Constraint (17) provides feasible regions for all variables.

Note here that the variable λ in constraint (9) records the value of the robust NF failure evaluation metric achieved by NF request through ξ_{st}^f. As the objective of the reformulation is to find the minimum λ, it also encourages evaluation metric value ξ_{st}^f to be minimized. Therefore, the above reformulation solves the maximal reliable NF deployment problem.

We next present the formulation for SFC service reliability.

4.4 | Formulations for SFC service reliability

Different from the non-chained NF failure probability, the SFC failure probability is

$$1 - \max_{\Gamma(F)} \min_{\substack{f \in F_{st} \\ d_{st} \in \mathcal{D}}} \max_{\eta_{st} \in \mathcal{P}_{st}} [1 - \Pi_{i \in \Gamma(f) \cap p_{st}} \rho_i] / \mid F_{st} \mid !$$

with $d_{st} \in \mathcal{D}$.

Proposition 22. *For requests with SFC, we have* $\max_{\Gamma(F)} \min_{\substack{f \in F_{st} \\ d_{st} \in \mathcal{D}}} \max_{p_{st} \in \mathcal{P}_{st}} [1 - \Pi_{i \in \Gamma(f) \cap p_{st}} \rho_i] / \mid F^*_{st} \mid ! = 1 - \min_{\Gamma(F)} \max_{\substack{f \in F_{st} \\ d_{st} \in \mathcal{D}}} \min_{p_{st} \in \mathcal{P}_{st}} \Pi_{i \in \Gamma(f) \cap p_{st}} \rho_i / \mid F^*_{st} \mid !$, *where* F^*_{st} *represents the requested NFs of*

$$d^*_{st} = \arg \min_{d_{st} \in \mathcal{D}, f \in F_{st}} [\Pi_{i \in \Gamma(f) \cap p_{st}} \rho_i].$$

We introduce here an auxiliary variable ω_{st} which indicates whether $d_{st} \in \mathcal{D}$ is selected as the d^*_{st}. By replacing routings from undirected to directed path set (i.e., $\eta_{st} \rightarrow p_{st}$) in constraints (12), (14), (16), (17), we present the formulation for the robust SFC provisioning as follows.

$$\min_{\lambda, \xi, \omega, \beta, y, x, z} \lambda$$

$$\begin{aligned}
\text{s.t. } & \lambda \geq \omega_{st}, && d_{st} \in \mathcal{D}, && (18)\\
& \omega_{st} \geq \xi^f_{st} - \ln \mid F_{st} \mid !, && f \in F,\ d_{st} \in \mathcal{D}, && (19)\\
& \sum_{d_{st} \in \mathcal{D}} \beta_{st} = 1, && && (20)\\
& \lambda \leq \omega_{st} + M(1 - \beta_{st}), && d_{st} \in \mathcal{D}, && (21)\\
& \lambda \geq \omega_{st} + M(\beta_{st} - 1), && d_{st} \in \mathcal{D}, && (22)\\
& \omega_{st} \geq 0, \beta_{st} \in \{0, 1\}, d_{st} \in \mathcal{D} && && (23)\\
& \text{Constraints (8) and (10)–(17)}
\end{aligned}$$

Constraint (18) is to guarantee the lower bound based on the $\mathcal{FP}$ (linearized). The newly introduced constraint (19) is used to capture the corresponding SFC request $d_{st} \in \mathcal{D}$. Constraint (20) guarantees that exactly one demand $d_{st} \in \mathcal{D}$ should be selected as the d^*_{st} which provides the $\mathcal{FP}(d_{st})$. Constraints (21) and (22) guarantee $\lambda = \omega_{st}$ for the selected d^*_{st} (when $\beta_{st} = 1$).

5 | SIMULATION RESULTS

We design our experiments for robust NF provisioning problems in two parts: (1) provisioning with non-chained NFs, and (2) provisioning with SFCs.

5.1 | Experiment design

5.1.1 | Design for robust NF provisioning with non-chained NFs

We select the NSF network as the physical network illustrated in Figure 5, which has 14 nodes and 21 links. NF requests are based on node pairs whose mappings onto physical nodes are known a priori. Six pairs of NF requests are constructed and listed as follows: (1, 2), (1, 4), (2, 3), (3, 5), (4, 7), and (6, 7). NF requests for logical arcs/links are randomly assigned with up to three NFs.

We consider that physical nodes have random failure probabilities, where the means of these probabilities are in the range of 1%-49% and the variance is 0.001. For each of the failure probabilities, we generate 25 testing samples and report their average as the results. For the simulations of the maximal reliable NF deployment problem, we first create testing cases which restrict the number of NF-enable nodes to be 40%, 50%, and 60% of the physical nodes.

Based on the settings above, two sets of testing cases are created. The first testing cases for the maximal NF reliable deployment problem have (i) NSF as the physical network, (ii) demands with up to three randomly assigned NF requests, (iii) a given limitation on the number of NF deployed locations, and (iv) random node failure probability. The proposed setting is to verify that when the number of NF locations decreases, whether the NF service reliability also goes down corresponding. Meanwhile, when the node failure probability increases, whether the NF service reliability also decreases.

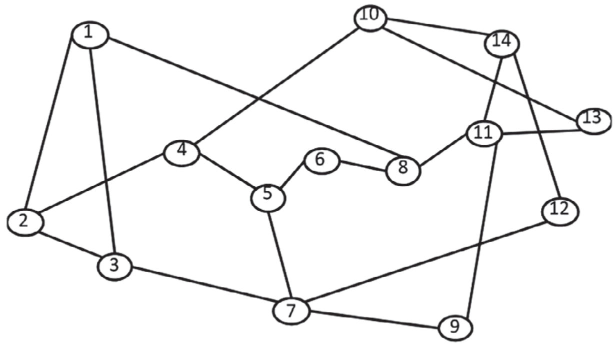

FIGURE 5 NSF

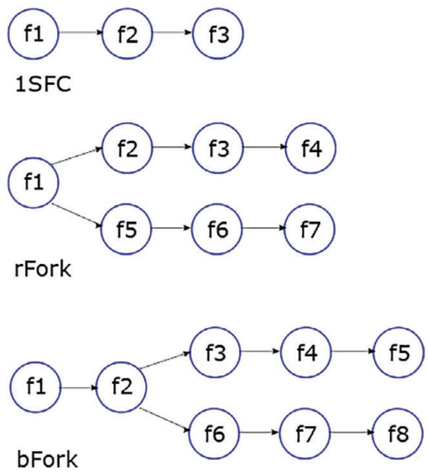

FIGURE 6 Forwarding graphs [Color figure can be viewed at wileyonlinelibrary.com]

The second testing cases have (i) a fixed NF service reliability (90%), and (ii) random physical node failure probability. The purpose of the setting is again to evaluate that with a fixed NF service reliability, whether extra NF-deployed nodes are required to fulfill the requirement of the service level when the node failure probability increases.

5.1.2 | Design for robust NF provisioning with SFCs

We consider three different forwarding graphs, the single SFC ("1SFC"), rooted fork ("rFork"), and branched fork ("bFork"), and illustrate them in Figure 6 to test the proposed robust evaluation metrics and approaches to calibrate their survivable probability. We take the CORONET network, illustrated in Figure 7, as the physical network which has 75 nodes, 99 links, and an average nodal degree of 2.6. We generate two sets of 6 demands and 10 demands randomly. For 1SFC as the forwarding graph, all demands require the SFC; for rFork and bFork, we randomly assign half of the demands (3 demands or 5 demands) with a branch of a fork, and the rest of the demands require the second branch of forks. Therefore, we have six testing cases, which are with three forwarding graphs and two demand pair setting correspondingly.

We also consider NF-enabled nodes to have failure probability from 1% to 50% with the variance be 0.1%. The experiment is designed to test with how many NF-enabled nodes, all demand pairs would have positive survivable probability and what the numerical values are. We report the robust survivability probability of all testing cases with different NF-enabled node failure probability.

5.2 | Computational results

5.2.1 | Computational results for robust NF provisioning with non-chained NFs

The simulation results for the maximal NF reliable deployment problem are presented in Figure 8. The three lines in blue, red, and green colors represent the testing cases with 40%, 50%, and 60% of NF-enabled physical nodes. The x-axis represents the physical node failure probability (in mean value) and the y-axis denotes the NF service reliability (in percentage). Each plotted node/dot in the figure presents the average NF service reliability for all testing samples. With up to 50% failure probability of the

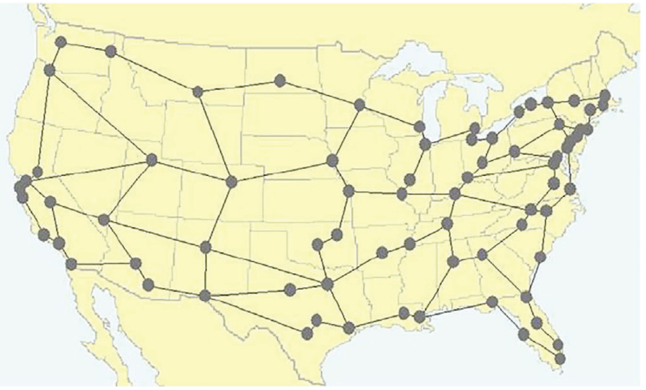

FIGURE 7 CORONET CONUS Network [Color figure can be viewed at wileyonlinelibrary.com]

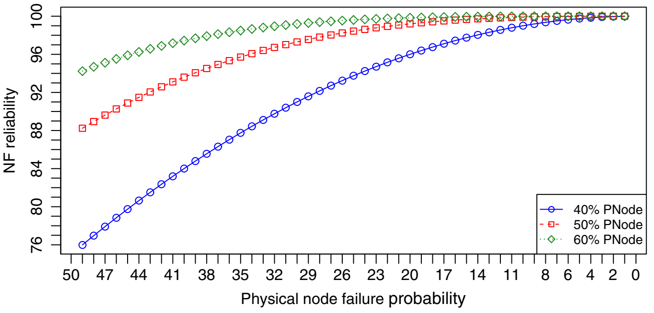

FIGURE 8 NF service reliability [Color figure can be viewed at wileyonlinelibrary.com]

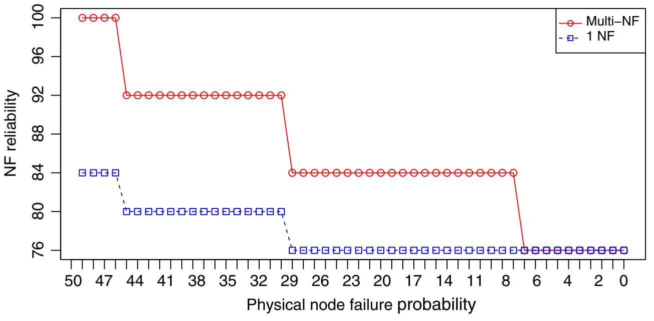

FIGURE 9 NF service reliability versus NF deployment [Color figure can be viewed at wileyonlinelibrary.com]

NF-enabled nodes, the NF reliability reaches 75%. When the number of NF-enabled nodes increases, the NF reliability increases to 87.5% and 93.7%, respectively. We confirm our analysis that with the limitation on the number of NF-enabled nodes, the NF service reliability increases when physical node failure probability decreases. Also, given the same physical node failure probabilities, we observe that when the number of NF-enabled nodes (in terms of the mean value) decreases, the reliability of the NF service decreases as well.

Figure 9 illustrates the number of NFs deployed to reach the required level of the NF service reliability (based on the maximal number of NF-enabled nodes in the testing cases) with single NF and multiple NFs (in our testing cases, three required NFs) in each demand. To reach the fixed (90%) NF service reliability, the number of physical nodes deployed with NFs is only doubled when the number of required NFs for each demand goes from one to three even with high failure probability (10%-50%) on physical nodes. The figure demonstrates a clear pattern between the number of nodes deployed with NFs and the NF service reliability.

In the simulation results, we observe that the NF service reliability is higher with more physical nodes deployed with the required NFs, and obviously, a lower average node failure probability leads to a higher NF service reliability under the failure(s) of physical nodes. The observations on these simulations are as expected and demonstrate the relationship between the number of NF-deployed nodes (cost-related restriction) and NF service reliability (service level).

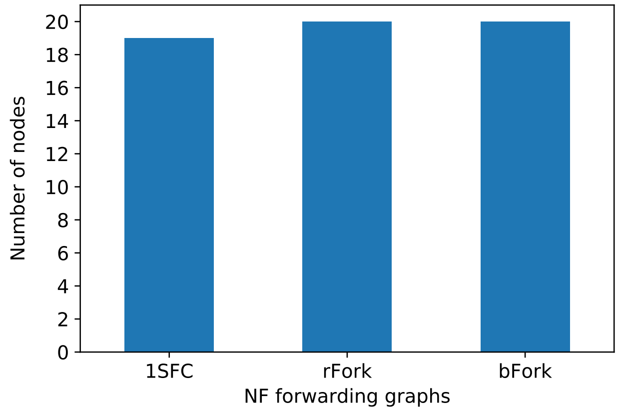

FIGURE 10 Number of NF-enabled nodes for six demand pairs [Color figure can be viewed at wileyonlinelibrary.com]

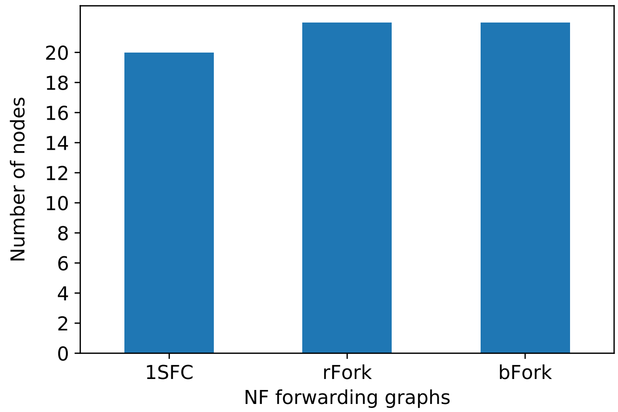

FIGURE 11 Number of NF-enabled nodes for 10 demand pairs [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 3 Robust survivable probability of SFC requests

failurePb	6Dm_1SFC	6Dm_rFork	6Dm_bFork	10dm_1SFC	10dm_rFork	10dm_bFork
5	94.1613	31.3871	6.2774	95.9597	29.8925	5.9785
10	93.4533	31.1511	6.2302	90.9091	28.3192	5.6638
15	92.2734	30.7578	6.1516	85.8586	26.7459	5.3492
20	90.6216	30.2072	6.0414	80.8081	25.1726	5.0345
25	88.4976	29.4992	5.8998	75.7575	23.5993	4.7199
30	85.9017	28.6339	5.7268	70.7073	22.0261	4.4052
35	82.8336	27.6112	5.5222	65.6568	20.4528	4.0906
40	79.2939	26.4313	5.2863	60.6062	18.8795	3.7759
45	75.2820	25.0940	5.0188	55.5557	17.3062	3.4612
50	70.7979	23.5993	4.7199	50.5051	15.7329	3.1466

5.2.2 | Computational results for robust NF provisioning with SFCs

Following the experiment design in Section 5.1.2, we test 6 SFC requests and 10 SFC requests cases separately.

Different from the robust survivable NF provisioning problem with non-chained NFs, the SFC requests need to visit required NFs in order. Therefore, the NF instance deployment on NF-enabled nodes is more restricted. We proceed with our testing in the larger-scale physical network, the CORONET network, to identify (1) how many NF-enabled nodes are needed, and (2) what are the corresponding survivable probabilities among all SFC requests to guarantee that all SFC requests are fulfilled with three types of SFC forwarding graphs.

We first report the NF-enabled nodes required to support all demands with different NF forwarding graphs, and have positive survivability, in Figures 10 and 11 for 6 and 10 SFC requests.

The results meet our expectation that with the same requests, more NF-enabled nodes are required to guarantee that all SFC requests have a positive survivable probability. As more requests are added, the requirement of NF-enabled nodes increased.

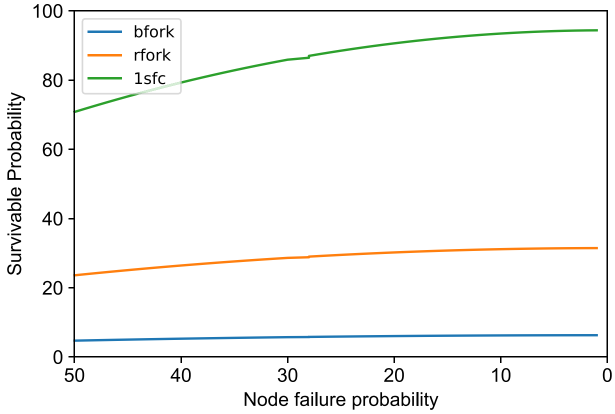

FIGURE 12 Six demand robust survivable probability [Color figure can be viewed at wileyonlinelibrary.com]

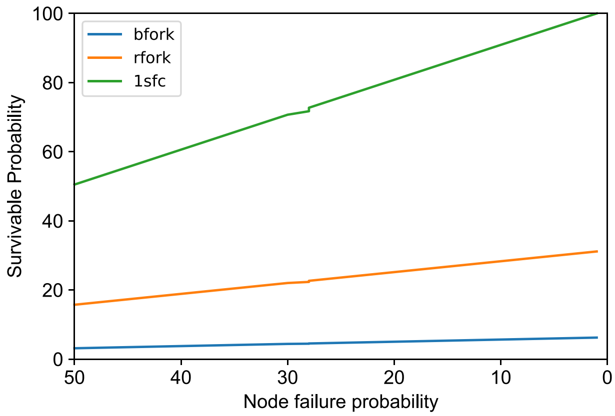

FIGURE 13 Ten demand robust survivable probability [Color figure can be viewed at wileyonlinelibrary.com]

Note here that we did not observe too much difference in the number of NF-enabled nodes needed with rFork and bFork as the NF forwarding graphs for both 6 and 10 SFC requests.

Next, we present the robust survivable probability of all testing cases with NF-enabled nodes failure probability from 5% to 50% (with a fixed 5% gap) for 6 and 10 demands in Table 3 and Figures 12 and 13. We let "failurePb," "6Dm_1SFC," "6Dm_rFork," "6Dm_bFork," "10dm_1SFC," "10dm_rFork," "10dm_bFork" represent the failure probability, survivable probability of 6 and 10 demands with 1SFC, rFork, and bFork as the forwarding graphs, respectively. Computational results show a very clear pattern that the higher the failure probability, the lower the survivable probability for all SFC requests. Compared with rFork and bFork, 1SFC as the forwarding graph has much higher survivable probability; and with bFork as the forwarding graph, the survivable probability is low. The highest we could reach is around 6.28%.

To have a finer granularity of the robust failure probability and observe the patterns of changes for robust survivable probability, we plot the robust failure probability in terms of NF-enabled failure probability from 50% to 1% (with 1% gap). With six SFC requests, the survivable probability is curved and convex. The trend line of the survivable probability has a smoother increase when the failure probability is lower. With 10 SFC requests, we observe that the changes to survivable probability are more linear in terms of the failure probability.

6 | CONCLUSION

In this study, we address the practical challenges in NFV 5G implementation. We propose a new robust evaluation metric that quantifies the minimal reliability among all NFs for all demands considering the random NF-enabled node failure. We study three sets of problems in the robust NF provisioning, that is, the SFC $s-t$ path problem, NFP-SFork, and NFP with the general NF forwarding graph. We introduce an auxiliary/augmented network layer and we develop pseudo-polynomial algorithms to solve the robust NF and SFC $s-t$ path problems. We present approximation algorithms for robust NFV with the SFC-Fork as the NF forwarding graph and adopt a two-step parameterized path reduction technique, which can serve in multiple types of approximation algorithms when the underlying network has the branching structure. Furthermore, we propose exact solution approaches via MILP with general forwarding graph structures. Computational results show that our proposed solution approaches are capable of managing robust NFP with non-chained NF and SFC requests in both small and large-size national-wide physical networks.

In further research, we would like to consider physical node capacity and NF deployment costs. We are also interested in evaluating the costs to introduce more NF-enabled nodes in the physical network. Another line of investigation is on the scenarios of shared risk group failure(s) and physical link failure(s) and their impacts on NF service reliability. Last, but not least, another research direction is to relax the assumptions on independent node failures, the correlation among NF-enabled node failures, and study their impacts on NF service reliability.

ORCID

Tachun Lin https://orcid.org/0000-0002-7193-3364

REFERENCES

[1] 5G-NORMA, *Definition and specification of connectivity and QoE/QoS management mechanisms*, Tech. report H2020-ICT-2014-2 5G NORMA, 5G NORMA, 2017.
[2] K. Aardal, F.A. Chudak, and D.B. Shmoys, *A 3-approximation algorithm for the* k*-level uncapacitated facility location problem*, Inform. Process. Lett. **72** (1999), 161–167.
[3] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati, *Network function virtualization in 5G*, IEEE Commun. Mag. **54** (2016), 84–91.
[4] A. Ageev, Y. Ye, and J. Zhang, *Improved combinatorial approximation algorithms for the k-level facility location problem*, SIAM J. Discrete Math. **18** (2004), 207–217.
[5] H.A. Alameddine, S. Sebbah, and C. Assi, *On the interplay between network function mapping and scheduling in VNF-based networks: A column generation approach*, IEEE Trans. Netw. Serv. Manage. **14** (2017), 860–874.
[6] N. Alliance, *5G white paper*, 2015.
[7] M.F. Bari, S.R. Chowdhury, R. Ahmed, and R. Boutaba, *On orchestrating virtual network functions*, 11th International Conference on Network and Service Management (CNSM), 2015, pp. 50–56.
[8] M.S. Bonfim, K.L. Dias, and S.F. Fernandes, *Integrated NFV/SDN architectures: A systematic literature review*, arXiv preprint arXiv:1801.01516, 2018.
[9] J. Byrka, "*An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem*," *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*Springer, Princeton, NJ, 2007, pp. 29–43.
[10] J. Byrka, M. Ghodsi, and A. Srinivasan, *LP-rounding algorithms for facility-location problems*, arXiv preprint arXiv:1007.3611, 2010.
[11] J. Byrka, S. Li, and B. Rybicki, *Improved approximation algorithm for* k*-level uncapacitated facility location problem (with penalties)*, Theory Comput. Syst. **58** (2016), 19–44.
[12] F. Carpio, S. Dhahri, and A. Jukan, *VNF placement with replication for load balancing in NFV networks*, IEEE International Conference on Communications (ICC), 2017, pp. 1–6.
[13] S. Chechik and D. Peleg, *Robust fault tolerant uncapacitated facility location*, arXiv preprint arXiv:0912.3188, 2009.
[14] S. Chechik and D. Peleg, *Robust fault tolerant uncapacitated facility location*, Theory Comput. Sci. **543** (2014), 9–23.
[15] G. Cheng, H. Chen, H. Hu, Z. Wang, and J. Lan, *Enabling network function combination via service chain instantiation*, Comput. Netw. **92** (2015), 396–407.
[16] N.M.K. Chowdhury, M.R. Rahman, and R. Boutaba, *Virtual network embedding with coordinated node and link mapping*, IEEE INFOCOM, 2009, pp. 783–791.
[17] R. Cohen, L. Lewin-Eytan, J.S. Naor, and D. Raz, *Near optimal placement of virtual network functions*, IEEE INFOCOM, 2015, pp. 1346–1354.
[18] S. D'Oro, L. Galluccio, S. Palazzo, and G. Schembra, *A game theoretic approach for distributed resource allocation and orchestration of softwarized networks*, IEEE J. Sel. Areas Commun. **35** (2017), 721–735.
[19] V. Eramo, M. Ammar, and F.G. Lavacca, *Migration energy aware reconfigurations of virtual network function instances in NFV architectures*, IEEE Access **5** (2017), 4927–4938.
[20] V. Eramo, E. Miucci, M. Ammar, and F.G. Lavacca, *An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures*, IEEE/ACM Trans. Netw. **25** (2017), 2008–2025.
[21] ETSI, *Network functions virtualisation (NFV): Management and orchestration*, Tech. report ETSI GS NFV-MAN 001, ETSI, 2014.
[22] ETSI, *Network functions virtualisation (NFV): Infrastructure overview*, Tech. report, ETSI, 2015.
[23] ETSI, *Network functions virtualisation (NFV); reliability; report on models and features for end-to-end reliability*, Tech. report V1.1.1, ETSI, 2016.
[24] ETSI, *Network functions virtualisation: White paper on NFV priorities for 5G*, Tech. report, ETSI, 2017.
[25] G. Even, M. Rost, and S. Schmid, *An approximation algorithm for path computation and function placement in SDNs*, International Colloquium on Structural Information and Communication Complexity, 2016, pp. 374–390.
[26] J. Fan, Z. Ye, C. Guan, X. Gao, K. Ren, and C. Qiao, *GREP: Guaranteeing reliability with enhanced protection in NFV*, ACM SIGCOMM Workshop on Hot Topics in Middleboxes and Network Function Virtualization, 2015, pp. 13–18.
[27] S. Guha and S. Khuller, *Greedy strikes back: Improved facility location algorithms*, J. Algorithms **31** (1999), 228–248.
[28] A. Gupta, B. Jaumard, M. Tornatore, and B. Mukherjee, *A scalable approach for service chain mapping with multiple SC instances in a wide-area network*, IEEE J. Sel. Areas Commun. **36** (2018), 529–541.
[29] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, *Network function virtualization: Challenges and opportunities for innovations*, IEEE Commun. Mag. **53** (2015), 90–97.
[30] B. Han, V. Gopalakrishnan, G. Kathirvel, and A. Shaikh, *On the resiliency of virtual network functions*, IEEE Commun. Mag. **55** (2017), 152–157.
[31] H. Hawilo, M. Jammal, and A. Shami, *Orchestrating network function virtualization platform: Migration or re-instantiation?* IEEE 6th International Conference on Cloud Networking (CloudNet), 2017, pp. 1–6.

[32] M. Jalalitabar, Service function graph design and embedding in next generation internet, Ph.D. thesis, Georgia State University, 2018.
[33] R. Krishnaswamy and M. Sviridenko, *Inapproximability of the multi-level uncapacitated facility location problem*, Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, 2012, pp. 718–734.
[34] S. Li, *A 1.488 approximation algorithm for the uncapacitated facility location problem*, Inform. Comput. **222** (2013), 45–58.
[35] T. Lin and Z. Zhou, *Robust virtual network function provisioning under random failures on network function enabled nodes*, Proc. of 10th International Workshop on Resilient Networks Design and Modeling (RNDM), 2018.
[36] T. Lin, Z. Zhou, M. Tornatore, and B. Mukherjee, *Demand-aware network function placement*, J. Lightwave Technol. **34** (2016), 2590–2600.
[37] R. Mijumbi, J. Serrat, J.L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, *Network function virtualization: State-of-the-art and research challenges*, IEEE Commun. Surv. Tutor. **18** (2016), 236–262.
[38] G. NFV-SWA, *001-V1.1.1-network functions virtualisation (NFV): Virtual network functions architecture*, 2014.
[39] N.C. Nguyen, P. Wang, D. Niyato, Y. Wen, and Z. Han, *Resource management in cloud networking using economic analysis and pricing models: A survey*, IEEE Commun. Surv. Tutor. **19** (2017), 954–1001.
[40] V.G. Nguyen, A. Brunstrom, K.J. Grinnemo, and J. Taheri, *SDN/NFV-based mobile packet core network architectures: A survey*, IEEE Commun. Surv. Tutor. **19** (2017), 1567–1602.
[41] L. Nobach, I. Rimac, V. Hilt, and D. Hausheer, *Statelet-based efficient and seamless NFV state transfer*, IEEE Trans. Netw. Serv. Manage. **14** (2017), 964–977.
[42] Openstack, *IETF SFC encapsulation*, 2018.
[43] A. Osseiran, F. Boccardi, V. Braun, K. Kusume, P. Marsch, M. Maternia, O. Queseth, M. Schellmann, H. Schotten, H. Taoka, *et al.*, *Scenarios for 5G mobile and wireless communications: The vision of the METIS project*, IEEE Commun. Mag. **52** (2014), 26–35.
[44] M. Peng, C. Wang, V. Lau, and H.V. Poor, *Fronthaul-constrained cloud radio access networks: Insights and challenges*, IEEE Wirel. Commun. **22** (2015), 152–160.
[45] L. Qu, C. Assi, K. Shaban, and M. Khabbaz, *Reliability-aware service provisioning in NFV-enabled enterprise datacenter networks*, International Conference on Network and Service Management (CNSM), 2016, pp. 153–159.
[46] M. Rost and S. Schmid, *Service chain and virtual network embeddings: Approximations using randomized rounding*, arXiv preprint arXiv:1604.02180, 2016.
[47] G. Sallam, G.R. Gupta, B. Li, and B. Ji, *Shortest path and maximum flow problems under service function chaining constraints*, arXiv preprint arXiv:1801.05795, 2018.
[48] D.B. Shmoys, É. Tardos, and K. Aardal, *Approximation algorithms for facility location problems*, Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing, 1997, pp. 265–274.
[49] H. Vardhan, S. Billenahalli, W. Huang, M. Razo, A. Sivasankaran, L. Tang, P. Monti, M. Tacca, and A. Fumagalli, *Finding a simple path with multiple must-include nodes*, 2009 IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems, 2009, pp. 1–3.
[50] W. Wang, Y. Liu, Y. Li, H. Song, Y. Wang, and J. Yuan, *Consistent state updates for virtualized network function migration*, IEEE Trans. Services Comput. (2017). https://ieeexplore.ieee.org/document/8081824
[51] Y. Wang, G. Xie, Z. Li, P. He, and K. Salamatian, *Transparent flow migration for NFV*, IEEE International Conference on Network Protocols (ICNP), 2016, pp. 1–10.
[52] L. Xia, Q. Wu, D. King, and H. Yokota, *VNF pool use cases*, Tech. report draft-king-vnfpool-mobile-use-case-02, IETF, 2014.
[53] Y. Xie, Z. Liu, S. Wang, and Y. Wang, *Service function chaining resource allocation: A survey*, arXiv preprint arXiv:1608.00095, 2016.
[54] M. Zeng, W. Fang, and Z. Zhu, *Orchestrating tree-type VNF forwarding graphs in inter-DC elastic optical networks*, J. Lightwave Technol. **34** (2016), 3330–3341.
[55] J. Zhang, *Approximating the two-level facility location problem via a quasi-greedy approach*, Math. Program. **108** (2006), 159–176.
[56] J. Zhang, W. Wu, and J. Lui, *On the theory of function placement and chaining for network function virtualization*, Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2018, pp. 91–100.

How to cite this article: Lin T, Zhou Z. Robust network function virtualization. *Networks.* 2020;1–25. https://doi.org/10.1002/net.21935

APPENDIX A

A.1 | Proof of Theorem 20

Lemma 23. *Given a feasible solution φ_{1fl} of the 1-level facility location problem based on NFP-SFork reduction and its corresponding NFP-SFork solution φ_{sfc}, we have $F(\varphi_{sfc}) + C(\varphi_{sfc}) \le F(\varphi_{1fl}) + C(\varphi_{1fl})$.*

Proof. Given a feasible solution $\varphi_{1\text{fl}}$ of the 1-level facility location problem, we have

$$C(\varphi_{1\text{fl}}) = \sum_{d \in D} [C(d,t) + C(p(t, i_{f_{b+1}}) + C(p(i_{f_{b+1}}, i_{f_1}))] + \sum_{t=1,\dots,|D|} F(p(t, i_{f_{b+1}})),$$

$$F(\varphi_{1\text{fl}}) = \sum_{i_{f_{b+1}} \in X_{b+1}} F(p(i_{f_{b+1}}, i_{f_1})). \tag{A1}$$

where X_{b+1} contains NF-enabled nodes selected by paths $\cup_{t\in 1,\ldots,|D|}p(t,i_{f_{b+1}})$. For the corresponding NFP-SFork solution φ_{sfc}, we have

$$\begin{aligned} O(\varphi_{\text{sfc}}) &\leq \sum_{i_{f_{b+1}}\in X_{b+1}} F(p(i_{f_{b+1}}, i_{f_1})),\\ C(\varphi_{\text{sfc}}) &= \sum_{d\in D}[C(d,t)+C(p(t,i_{f_{b+1}}))+C(p(i_{f_{b+1}},i_{f_1}))]+\sum_{t=1,\ldots,|D|}F(p(t,i_{f_{b+1}})). \end{aligned} \tag{A2}$$

Hence, the conclusion holds. ■

Lemma 24. *Given a solution of NFP-SFork, χ_{sfc}, then there exists another solution ψ_{sfc} such that*

1. *in solution χ_{sfc} and given $\lambda\in\Lambda$, paths $\rho_1=(i_{f_{1(\lambda)}},i_{f_{2(\lambda)}},\ldots,i_{f_{\gamma(\lambda)}})$ and $\rho_2=(i'_{f_{1(\lambda)}},i'_{f_{2(\lambda)}},\ldots,i'_{f_{\gamma(\lambda)}})$ with $\ell(\lambda)$ for some $i_{\ell(\lambda)}=i'_{\ell(\lambda)}$, then in solution ψ_{sfc}, $i_{j(\lambda)}=i'_{j(\lambda)}$ with $1\leq j(\lambda)\leq\ell(\lambda)$, and*
2. *in solution χ_{sfc} with given λ, $\lambda'\in\Lambda$, paths $\rho_1=(i_{f_{1(\lambda)}},i_{f_{2(\lambda)}},\ldots,i_{f_{|\lambda|}})$ and $\rho_2=(i'_{f_{1(\lambda')}},i'_{f_{2(\lambda')}},\ldots,i'_{f_{|\lambda'|}})$ with $i_{f_{\ell(\lambda)}}=i'_{f_{\ell'(\lambda')}}$ for some $\ell(\lambda)$, $\ell'(\lambda')$, then in solution ψ_{sfc}, $i_j=i'_j$, for all $1\leq j\leq\ell(\lambda)$; and*
3. *$O(\psi_{sfc})\leq O(\chi_{sfc})$ and $C_{|\lambda|}(\psi_{sfc})=C_{|\lambda|}(\chi_{sfc})$, and $\sum_{1\leq j\leq|\lambda|-1}C_j(\psi_{sfc})\leq\sum_{1\leq j\leq|\lambda|-1}C_j(\chi_{sfc})$.*

Proof. *Proof of Claim* 1: Given a solution χ_{sfc} of NFP-SFork, we show how to obtain ψ_{sfc} based on χ'_{sfc}, a feasible solution for NFP-SFork. In χ_{sfc}, there exist paths $\rho_1=(i_{f_{1(\lambda)}},i_{f_{2(\lambda)}},\ldots,i_{f_{\gamma(\lambda)}})$ and $\rho_2=(i'_{f_{1(\lambda)}},i'_{f_{2(\lambda)}},\ldots,i'_{f_{\gamma(\lambda)}})$ with ℓ_λ for some $i_{f_{\ell(\lambda)}}=i'_{f_{\ell(\lambda)}}$, but for all $1\leq h(\lambda)\leq\ell_\lambda$, $\rho_1(f_{h(\lambda)})\neq\rho_2(f_{h(\lambda)})$. We let $\rho_2(f_{\ell(\lambda)})=\rho_1(f_{\ell(\lambda)})$, and only deploy VNF instances on ρ_1 without the deployment on $\rho_2(f_{\ell(\lambda)})$, which is still a feasible solution for NFP-SFork. In ψ_{sfc}, $\rho_1=(i_{f_{1(\lambda)}},i_{f_{2(\lambda)}},\ldots,i_{f_{\ell(\lambda)}},\ldots,i_{f_{\gamma(\lambda)}})$ and $\rho_2=(i'_{f_{1(\lambda)}},i'_{f_{2(\lambda)}},\ldots,i_{f_{\ell(\lambda)}},\ldots,i'_{f_{\gamma(\lambda)}})$ with ℓ_λ.

Proof of Claim 2. Similar to the proof above, with solution χ_{sfc}, we alter the SFC service path ρ_2 as

$$\rho_2=(i_{f_{1(\lambda)}},i_{f_{2(\lambda)}},\ldots,i_{f_{\ell(\lambda)}},\ldots,i'_{f_{|\lambda'|}})),$$

which still provides a feasible solution for NFP-SFork. Solution ψ_{sfc} takes ρ_1 and alters ρ_2 as the service paths.

With Claims 1 and 2, Claim 3 holds. ■

Lemma 25. $\alpha F(\psi_{1fl})+\beta C(\psi_{1fl})\leq\alpha O(\psi_{sfc})+3\beta C(\psi_{sfc})$.

Proof. Given a solution χ_{sfc}, we construct ψ_{sfc}. Without loss of generality, we assume that ψ_{sfc} satisfies Lemma 17. Thus, we have $O(\psi_{\text{sfc}})\leq O(\chi_{\text{sfc}})$.

We let $X_1\in N(f_1)$ and $X_{b+1}\in N(f_{b+1})$ be two NF-enabled node sets to be deployed with NF instances. With Lemma 24, we let $D(\mu)\subset D$ be the request set connecting to $\mu\in X_{b+1}$, where $p(\mu)=\arg\min_{d\in D(\mu)}C(p(d,i_{f_{b+1}}))$. We let $I(v)\in N(f_{b+1})$ be NF b-enabled nodes connecting to $v\in X_1$ following the solution χ_{sfc}. We let

$$p(v)=\arg\min_{i_{f_{b+1}}\in I(f_{b+1})}C(p(i_{f_{b+1}},i_{f_1})),$$

$\mathcal{P}_1$ be the SFC service subpath set in ψ_{sfc} connecting NF 1 enabled nodes to the NF b enabled nodes; and $\mathcal{P}_2$ be the SFC service subpath set in ψ_{sfc} connecting NF $(b+1)$ enabled node to NF $|\gamma(\lambda)|$ enabled node. We create a new solution, ς_{sfc}, for NFP-SFork where all demands in $D(\mu)$ are connected to $p(\mu)$, and $j\in I(f_{b+1})(v)$ are connected to $p(v)$.

Hence, we decompose the connection cost of NFP-SFork into four parts:

$$\begin{aligned} C(\psi_{\text{sfc}}) &= C_\gamma(\psi_{\text{sfc}})+C_b(\psi_{\text{sfc}})+C_{\mathcal{P}_1}(\psi_{\text{sfc}})+C_{\mathcal{P}_2}(\psi_{\text{sfc}})\\ \text{where}\ \ C_\gamma(\psi_{\text{sfc}}) &= \sum_{(d,i_{f_{\gamma(\lambda)}})\in\bigcup_{d\in D}p(d,i_{f_{b+1}})} C(d,i_{f_{\gamma(\lambda)}}),\\ C_b(\psi_{\text{sfc}}) &= \sum_{(i_{f_{b+1}},i_{f_b})\in\bigcup_{i_{f_{b+1}}\in X_{b+1}}p(i_{f_{b+1}},i_{f_1})} C(i_{f_{b+1}},i_{f_b}),\\ C_{\mathcal{P}_1}(\psi_{\text{sfc}}) &= \sum_{i_{f_{b+1}}\in X_{b+1}} C(p(i_{f_{b+1}},i_{f_1})),\\ C_{\mathcal{P}_2}(\psi_{\text{sfc}}) &= \sum_{t=1,\ldots,|D|} C(p(t,i_{f_{b+1}})). \end{aligned}$$

For $d\in D(\mu)$ with $\mu\in X_b$,

$$C_{\mathcal{P}_2}(\varsigma_{\text{sfc}})+C_\gamma(\varsigma_{\text{sfc}})\leq 3C_{\mathcal{P}_2}(\psi_{\text{sfc}})+C_\gamma(\psi_{\text{sfc}})$$

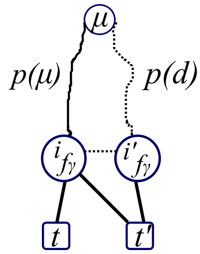

FIGURE A1 Triangle inequality for connection costs [Color figure can be viewed at wileyonlinelibrary.com]

With the fact that given a demand $d \in D(\mu)$, $C(p(u), \varsigma_{\text{sfc}}) + C((d, p(u)), \varsigma_{\text{sfc}}) \leq 3C(p(d), \psi_{\text{sfc}}) + C((d, p(u)), \psi_{\text{sfc}})$ follows the triangle inequality (as illustrated in Figure A1). Through a similar idea, we have

$$C_{\mathcal{P}_1}(\varsigma_{\text{sfc}}) + C_{(b,b+1)}(\varsigma_{\text{sfc}}) \leq 3C_{\mathcal{P}_1}(\psi_{\text{sfc}}) + C_{(b,b+1)}(\psi_{\text{sfc}}).$$

We create the 1-level facility location solution, $\varsigma_{1\text{fl}}$, as follows: all NF-enabled nodes on path $p(v, |I(v)|)$ and $p(j, |D_\mu|)$ with $j \in X_{b+1}$ are deployed with NFs, and d is connected to path $p(j, |D_\mu|)$ where $d \in D_\mu$, and $p(j, |D(\mu)|)$ is connected to $p(v, |I(v)|)$, where $j \in I(v)$.

$$\begin{aligned}
&\alpha O(\varsigma_{1\text{fl}}) + \beta C(\varsigma_{1\text{fl}}) \\
&\quad = \alpha \sum_{v \in X_1} O(p(v, |I(v)|)) + \beta \sum_{v \in X_1} \sum_{j \in I(v)} C(p(v, |I(v)|)) + \beta C_b \\
&\qquad + \beta \sum_{\mu \in X_{b+1}} \sum_{d \in D(\mu)} [C(p(d, \mu)) + O(p(d, \mu))] + \beta C_\gamma \\
&\quad \leq \sum_{v \in X_1} [\alpha O(p(v, |I(v)|)) + \beta |I(v)| C(p(v, |I(v)|))] + \beta C_b \\
&\qquad + \sum_{\mu \in X_{b+1}} \beta |D(\mu)| [C(p(d, \mu)) + O(p(d, \mu))] + \beta C_\gamma \\
&\quad \leq \sum_{v \in X_1} [\alpha O(p(v)) + \beta |I(v)| C(p(v))] + \beta C_b \\
&\qquad + \sum_{\mu \in X_{b+1}} |D(\mu)| [\beta O(p(\mu)) + \beta C(p(\mu))] + \beta C_\gamma \\
&\quad \leq \alpha O_1(\varsigma_{\text{sfc}}) + \beta [C_{\mathcal{P}_1}(\varsigma_{\text{sfc}}) + C_b(\varsigma_{\text{sfc}}) + C_{\mathcal{P}_2}(\varsigma_{\text{sfc}}) \\
&\qquad + C_\gamma(\varsigma_{\text{sfc}}) + O_2(\varsigma_{\text{sfc}})],
\end{aligned} \tag{A3}$$

where inequality (A3) holds with two-step parameterized path reduction.

Hence, the conclusion holds. ■