DOI:
10.1109/JSAC.2018.2815318

Document Version
Peer reviewed version

Link to publication record in King's Research Portal

vSPACE: VNF Simultaneous Placement, Admission Control and Embedding

Mohammad Ali Tahmasbi Nejad, Saeedeh Parsaeefard, Mohammad Ali Maddah-Ali, Toktam Mahmoodi, and Babak Hossein Khalaj

***Abstract*—In future wireless networks, network functions virtualization (NFV) lays the foundations for establishing a new dynamic resource management framework to efficiently utilize network resources. In this context, a network service can be viewed as a chain of virtual network functions (VNFs), called a service function chain (SFC), served via placement, admission control (AC), and embedding into network infrastructure, based on the resource management objectives and the state of network. To fully exploit such a potential and reach higher network performance, resource management stages should be jointly performed. To this end, two main challenges are: how to present a system model that formulates the desired resource allocation problem for different types of SFCs as well as different features, and how to tackle the computational complexity of the problem and solve it in a tractable manner. In this paper, we address these two issues and solve the joint problem of AC and SFC embedding. We introduce a comprehensive system model, and formulate the joint task as a mixed integer linear programming (MILP). This formulation encompasses *splittable VNF* and *multi-path routing* scenarios. We employ relaxation, reformulation, and successive convex approximation (SCA) methods to solve the problem. Simulation results demonstrate that the proposed schemes outperform the earlier works.**

***Index Terms*—Network functions virtualization (NFV), resource allocation, virtual network embedding, virtual network function (VNF) placement.**

I. Introduction

The advent of a large number of information and communications technology (ICT) services as well as the quality requirements of the fifth generation mobile networks (5G) necessitate the use of pioneering technologies such as software-defined networking (SDN) and network functions virtualization (NFV), with a higher level of flexibility and agility [1], [2].

A certain service provided by an operator can be viewed as a chain of network functions, called a service function chain (SFC) [3]. Such an SFC is identified by the number and types of network functions included as well as their arrangement and the chronological order in which they need to be carried out. Inspired by the fundamental idea behind NFV, i.e. decoupling the software implementation of network functions from the hardware infrastructure, the network functions in NFV are no longer device-oriented, and can be deployed on general-purpose hardware rather than proprietary dedicated hardware. These virtual network functions (VNFs) can be easily installed in/removed from a server, or migrate from one server to another in order to accomplish the service requests' objectives in compliance with the requirements stipulated by service level agreement (SLA) [4]–[6]. We refer to these network service requests, each equivalent to a chain of VNFs, as VNF requests (VNFRs).

As proposed by European Telecommunications Standards Institute (ETSI), NFV architecture is composed of three main components: services, NFV infrastructure (NFVI), and the NFV management and orchestration (NFV-MANO) [7]. Services should be implemented on the physical and virtual resources provided in the substrate network, or NFVI. NFV-MANO undertakes the management tasks such as VNFs configuration and provisioning, based on the knowledge acquired from its data repositories and other components' states.

Within such a framework, one of the main problems to be considered is the problem of NFV resource allocation. This issue encompasses a number of aspects such as [8]: 1) VNFs chain composition (VNFs-CC), i.e. dealing with efficient ordering of a service's VNFs [9], [10]; 2) VNF forwarding graph embedding (VNF-FGE), i.e. seeking to embed VNFs in the substrate network, a problem commonly divided into two sub-problems of virtual node mapping and virtual link mapping [11]–[14]; and 3) VNFs scheduling (VNFs-SCH), i.e. time ordering the execution of VNFs, a problem arising when a resource, e.g. a server, is shared among several VNFs of different VNFRs over multiple time-slots and the time-slots allocated to execute each VNF should be determined, with some objectives such as total execution time minimization [15], [16].

Along with these three major aspects of NFV resource allocation, the other important tasks that can effectively improve the performance of NFV networks are VNFRs admission control (AC) and re-composition of SFCs [8]. AC is the process of deciding whether to accept a VNFR to be served in the network or reject it, with a set of aims such as complying with the constraints on network resources, meeting VNFRs' requirements, and optimizing some objective functions. Furthermore, through SFC re-composition stage,

Mohammad Ali Tahmasbi Nejad is with Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran, and Department of Communication Technologies, Iran Telecommunication Research Center (ITRC), Tehran, Iran (e-mail: m_tahmasbi@ee.sharif.edu).

Saeedeh Parsaeefard is with Department of Communication Technologies, Iran Telecommunication Research Center (ITRC), Tehran, Iran (e-mail: s.parsaeifard@itrc.ac.ir).

Mohammad Ali Maddah-Ali is with Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran (e-mail: maddah_ali@sharif.edu).

Toktam Mahmoodi is with Centre for Telecommunications Research, Department of Informatics, King's College London, London, U.K. (e-mail: toktam.mahmoodi@kcl.ac.uk).

Babak Hossein Khalaj is with Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran (e-mail: khalaj@sharif.edu).

VNFs can be added, removed, or re-ordered, based on the correlation between SFCs as well as the network's state, in order to enhance the performance of resource allocation.

In addition, performing any set of the NFV resource allocation stages in a coordinated way has a significant impact on performance improvement. Thus, NFV resource allocation should address the following two key issues: how to present a comprehensive system model to jointly formulate the different aforementioned aspects, and how to deal with the NP-hardness of resulting optimization problems. In order to address such issues, within the scope of the present study, we investigate the problems of virtual node/link mapping (that is, the two sub-problems of VNF-FGE, namely virtual node and link mappings, are considered and solved simultaneously) and AC in a joint manner. The main contribution of this paper is to propose a general system model to formulate the joint problem of AC and VNF-FGE – referred to as AC/FGE in the sequel – and to solve this NP-hard problem (refer to [17] for the proof of NP-hardness) in a tractable manner.

It should be emphasized that considering the joint problem of AC/FGE is of great importance. This is because, despite the great deal of attention devoted to VNF-FGE and other stages of NFV resource allocation in the literature, AC problem has been rather neglected in this context. Moreover, AC/FGE not only introduces a joint problem formulation, but also provides a joint solution to handle the two stages together. Nonetheless, other works studying AC generally neither present a comprehensive joint formulation nor take joint approaches to solve the problem. As a result, either sequentially handling the requests in an online manner [18], [19] or separating AC process from VNF-FGE [20] blemishes the optimality of the solution. Therefore, it is readily observable that AC/FGE can significantly affect the outcome of the process and provide a higher level of performance employing a slightly-studied aspect of NFV resource allocation problem.

In this paper, we consider an NFV-enabled network, where the VNFRs, at each time-slot, are disclosed to the NFV-MANO by their related SFCs. The AC/FGE problem is considered, where not only are the decisions of admitting and embedding a set of requests made in a joint manner, but also the two sub-problems of VNF-FGE, i.e. virtual node/link mapping, are jointly optimized.

We also propose a comprehensive system model to represent services, through introduction of *dummy nodes*, which provides the capability to formulate AC/FGE problem with only linear constraints. A dummy node is a zero-capacity virtual node added to the original SFC of VNFRs, which is aimed at reducing the bandwidth consumption in some scenarios such as postponing the data splitting in a multicast data transmission. Such a system model, obtained through proper introduction of dummy nodes, is of critical value in scenarios such as network caching, multiple input/output services, and correlated data flows.

Our formulation also provides the possibility of taking the cases of *splittable VNF* and *multi-path routing*. The splittable VNF case describes a situation where VNFs operate at packet level rather than flow level. Particularly, several loosely coupled components of a certain VNF in SFC can be instantiated on different servers, each serving a sub-flow of the original flow. In addition, multi-path routing represents the scenarios where the flow between two subsequent VNFs can be routed through multiple paths, and out-of-order delivery is tolerable.

Therefore, two variants of AC/FGE problem are investigated: 1) *hard AC/FGE*, subject to no VNF splitting or multi-path routing, which is formulated by an integer linear programming (ILP), and 2) *soft AC/FGE*, where VNF splitting and multi-path routing are permissible, formulated by a mixed integer linear programming (MILP). To cope with such NP-hard problems, we introduce a successive convex approximation (SCA) method for hard AC/FGE and a heuristic approach for soft AC/FGE. The simulation results clearly demonstrate the advantage of the proposed AC/FGE approach in comparison with earlier works in terms of network's revenue.

The remainder of the paper is organized as follows. Related work is presented in Section II. Section III describes the system model. In Section IV, the problem formulation of AC/FGE is introduced, as an ILP and a MILP in the cases of hard and soft AC/FGE, respectively. Next, in Section V, an SCA approach is proposed to tackle the hard AC/FGE problem, where VNF splitting and multi-path routing are prohibited. In Section VI, we introduce a heuristic method for soft AC/FGE problem, which allows adopting splittable VNF and multi-path routing. Section VII is devoted to presenting numerical illustrations, and finally Section VIII concludes the paper.

II. Related Work

VNF-FGE problem has recently received a great deal of attention in the literature. This problem may be classified from different aspects such as objectives, system models, and solutions to tackle the computational complexity of the problems. For instance, a number of works have formulated VNF-FGE as integer programming problems, attempting to tackle the problem by employing greedy or heuristic algorithms [11], [21]–[25]. Bari et al. in [11] provide a MILP formulation for VNF-FGE to optimize network operational costs and utilization through implementation in CPLEX, and also propose a dynamic programming-based heuristic. The authors of [21] also introduce an ILP model as well as a heuristic to minimize the number of deployed VNF instances.

From the system model's perspective, in [22], a hybrid scenario has been considered, where some services are provided by dedicated physical hardware, while the rest are treated as SFCs, and served by software instances. In this NFV environment, Moens and De Turck present an ILP with the objective of minimizing the number of used servers, implemented through CPLEX for model evaluation. Aiming to minimize the number of cores utilized by VNF instances, an ILP has been provided by Addis et al. [23], eliciting tradeoffs between the goals of traffic engineering in legacy and NFV networks.

With respect to the objective of resource allocation problem, a number of other works have considered VNF-FGE problem with objectives such as energy efficiency, service resilience, and load balancing [26]–[29]. In [26], a game theory-based solution for resource allocation in NFV has been provided, which aims at minimizing the product of energy consumption and processing delay. In [30], a game-theoretic approach is adopted to investigate the Nash equilibrium solution by formulating the VNFs-CC problem as a congestion game.

Another group of related works consider solving more than one stage of NFV resource allocation [10], [15], [31], [32]. For instance, the scheme proposed in [10] employs a greedy algorithm to solve VNFs-CC, which provides the input to VNF-FGE stage, subsequently solved as a quadratically constrained program. In contrast to [10], coordinated solutions have been introduced in [31] and [15] to deal with VNFs-CC and VNFs-SCH respectively, in addition to VNF-FGE.

The AC problem, on the other hand, has not been the focus of much earlier studies in NFV context. [20] considers the problems of AC and VNF-FGE, and uses a greedy algorithm, where the sorted requests are dealt with in a certain order to discover whether they can be embedded in the substrate network. Some other works also assume the requests arrive sequentially, and the AC decision needs to be made on each demand's arrival [18], [19]. In the present study, however, AC and VNF-FGE problems are jointly considered to ensure that the decisions of admitting and embedding of a set of requests are made in a coordinated manner.

III. System Model

We consider a network provider which receives VNFRs to be served over a substrate network. We assume that requests' arrival and departure occur in a time-slotted manner, so that, at the end of each time-slot, the VNFRs whose tasks have already been completely carried out are evicted, and the ones which have been chosen to be served will commence execution at the beginning of the following time-slot. The notations used to model such scenarios are provided in Table I. Also, some other notations used throughout the paper are as follows.

Notations: $|\cdot|$ returns either the absolute value of a real number, or the cardinality of a set. $\|\cdot\|_2$ denotes the ℓ_2-norm of a vector, and $(\cdot)^T$ represents transpose operation. $\mathbf{0}_{m,n}$, $\mathbf{1}_{m,n}$, and $\boldsymbol{I}_n$, for any $m, n \in \mathbb{N}^+$, are the $m \times n$ null matrix, the $m \times n$ all-ones matrix, and the identity matrix of size n, respectively (if the size can be trivially determined by the context, the subscript may be eliminated).

A. NFVI

To demonstrate the state of NFVI at the beginning of each time-slot, we consider a substrate network comprised of N nodes which interconnect through a number of F links, each of a given bandwidth. This network can be modeled as an undirected graph $\bar{\mathcal{G}} = (\bar{\mathcal{V}}, \bar{\mathcal{E}}, \bar{\mathcal{C}}, \bar{\mathcal{B}})$, with $\bar{\mathcal{V}} = \{\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_N\}$ and $\bar{\mathcal{E}} = \{\bar{e}_1, \bar{e}_2, \ldots, \bar{e}_F\}$ denoting the vertices and the edges of the graph, respectively. $\bar{\mathcal{C}} = \{\bar{\boldsymbol{c}}_1, \bar{\boldsymbol{c}}_2, \ldots, \bar{\boldsymbol{c}}_N\}$ consists of the capacities of vertices, and the maximum data rate at which data can be carried between each pair of nodes is embodied in $\bar{\mathcal{B}} = \{\bar{b}_1, \bar{b}_2, \ldots, \bar{b}_F\}$.

NFVI provides different computing, storage, and networking resources to host VNFs [7]. Based on the resource types offered in the network, each request's requirements of physical resources can be expressed as a vector whose size is in compliance with the number of categories into which NFVI resources are divided. So, if there are S different types of resources available in the network, a vector $\bar{\boldsymbol{c}}_n = \big(\bar{c}_n(1), \bar{c}_n(2), \ldots, \bar{c}_n(S)\big)$ determines the amount of each resource type available in a certain substrate node.

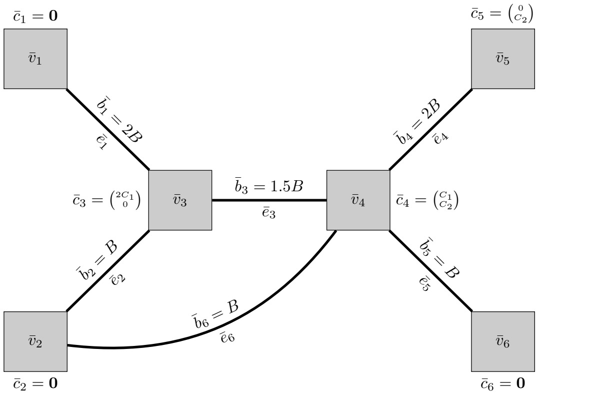


Fig. 1: A simple illustration of a substrate network with 6 nodes, 3 of which are capable of hosting VNFs, categorized into 2 different types (for fixed scalars C_1, C_2, and B, each with its corresponding dimension).

It should be noted that, in the present model of NFVI, pure forwarding devices and general-purpose servers do not need to be considered as distinct set of nodes. In other words, the NFVI graph is composed of the nodes of all types in the substrate network, where a capacity vector of $\mathbf{0}$ is assigned to pure forwarding devices in order to show that they are not able to host VNFs.

Therefore, to obtain the substrate network graph from NFVI, every network device is represented by a node, together constituting $\bar{\mathcal{V}}$. Next, an edge should be considered in $\bar{\mathcal{E}}$ for each pair of interconnected devices, whose capacity equals the link bandwidth connecting the two nodes and is included in $\bar{\mathcal{B}}$. Then, the number of different resources offered by NFVI to host VNFs (such as CPU, RAM, disk, and NIC [8]) needs to be determined as S. Finally, a capacity vector of size S is assigned to every substrate node, whose entries are initialized according to the amount of corresponding resources available in the node. Also, the devices which are not NFV-enabled and not able to deploy VNFs, such as physical switches and routers, are considered to have $\mathbf{0}$ capacity. These capacities as $\bar{\mathcal{C}}$, along with the aforementioned sets, form the substrate graph $\bar{\mathcal{G}}$.

An example of such a substrate network has been illustrated in Fig. 1. In this example, two types of resources are assumed, leading to bi-dimensional capacity vectors for substrate nodes. As the capacity of $\mathbf{0}$ implies, $\bar{v}_1$, $\bar{v}_2$, and $\bar{v}_6$ are pure forwarding devices. On the other hand, $\bar{v}_3$, $\bar{v}_5$, and $\bar{v}_4$ are able to host VNFs of type 1, type 2, and both types, respectively.

B. Services

We denote by $\mathcal{R} = \{R_1, R_2, \ldots, R_K\}$ the set of all requests which await authorization to be served and need to be handled at the beginning of the current time-slot—including VNFRs

TABLE I: Nomenclature

Symbol	Description
	NFVI
$\bar{\mathcal{G}}$	The graph modeling the NFVI
$\bar{\mathcal{V}}$	$\bar{\mathcal{G}}$ substrate nodes
$\bar{\mathcal{E}}$	$\bar{\mathcal{G}}$ substrate links
$\bar{\mathcal{C}}$	$\bar{\mathcal{V}}$ corresponding available capacities
$\bar{\mathcal{B}}$	$\bar{\mathcal{E}}$ corresponding available bandwidth
$\bar{v}_n$	n-th substrate node in $\bar{\mathcal{V}}$
$\bar{e}_f$	f-th substrate link in $\bar{\mathcal{E}}$
$\bar{\boldsymbol{c}}_n$	$\bar{v}_n$ available capacity vector
$\bar{b}_f$	$\bar{e}_f$ available bandwidth
S	Number of different available types of resources
$\bar{c}_n(s)$	$\bar{v}_n$ available capacity of type s
N	$\bar{\mathcal{V}}$ cardinality
F	$\bar{\mathcal{E}}$ cardinality
$\bar{\mathcal{P}}$	Set of available paths in $\bar{\mathcal{G}}$
$\bar{P}_j$	j-th path in $\bar{\mathcal{P}}$
$\bar{v}_{n_{j,q}}$	q-th node in $\bar{P}_j$
$\bar{e}_{f_{j,q}}$	q-th link in $\bar{P}_j$
Q_j	Number of nodes included in $\bar{P}_j$
J	$\bar{\mathcal{P}}$ cardinality
$\hat{P}_n$	Internal path in substrate node n
	VNFRs
$\mathcal{R}$	Set of all VNFRs
R_k	k-th request in $\mathcal{R}$
K	Total Number of VNFRs
$\mathcal{G}^k$	The graph representing k-th VNFR
$\mathcal{V}^k$	$\mathcal{G}^k$ virtual nodes
$\mathcal{E}^k$	$\mathcal{G}^k$ virtual links
$\mathcal{C}^k$	$\mathcal{V}^k$ corresponding demanded capacities
$\mathcal{B}^k$	$\mathcal{E}^k$ corresponding demanded bandwidth
$\mathcal{L}^k$	$\mathcal{V}^k$ corresponding allowed embedding locations
v_m^k	m-th virtual node in $\mathcal{V}^k$
e_i^k	i-th virtual link in $\mathcal{E}^k$
$\boldsymbol{c}_m^k$	v_m^k demanded capacity vector
b_i^k	e_i^k demanded bandwidth
ℓ_m^k	v_m^k allowed embedding locations
$c_m^k(s)$	v_m^k demanded capacity of type s
M_k	$\mathcal{V}^k$ cardinality
I_k	$\mathcal{E}^k$ cardinality
M	Total number of virtual nodes in $\mathcal{R}$
I	Total number of virtual links in $\mathcal{R}$
	AC/FGE variables
$\boldsymbol{X}$	Virtual node mapping $M \times N$ matrix
$\boldsymbol{X}^k$	The $M_k \times N$ sub-matrix taken out of $\boldsymbol{X}$ and related to k-th request's virtual nodes
$x_{m,n}^k$	The entry of $\boldsymbol{X}^k$ at m-th row and n-th column
$\boldsymbol{Y}$	Virtual link mapping $I \times J$ matrix
$\boldsymbol{Y}^k$	The $I_k \times J$ sub-matrix taken out of $\boldsymbol{Y}$ and related to k-th request's virtual links
$y_{i,j}^k$	The entry of $\boldsymbol{Y}^k$ at i-th row and j-th column
$\hat{\boldsymbol{Y}}$	Internal link mapping $I \times N$ matrix
$\hat{\boldsymbol{Y}}^k$	The $I_k \times N$ sub-matrix taken out of $\hat{\boldsymbol{Y}}$ and related to k-th request's virtual links
$\hat{y}_{i,n}^k$	The entry of $\hat{\boldsymbol{Y}}^k$ at i-th row and n-th column
$\boldsymbol{z}$	AC/FGE optimization variable
$\boldsymbol{r}$	Revenue vector
r_k	The revenue made by serving k-th request
$\boldsymbol{\alpha}$	Admission vector
α_k	The binary variable representing whether k-th request is admitted

having arrived during past time-slot as well as earlier ones not handled yet.

Each requested service chain R_k is identified by a virtual directed graph $\mathcal{G}^k = (\mathcal{V}^k, \mathcal{E}^k, \mathcal{C}^k, \mathcal{B}^k, \mathcal{L}^k)$. The set of virtual nodes of the k-th request is denoted by $\mathcal{V}^k = \{v_1^k, v_2^k, \ldots, v_{M_k}^k\}$. The corresponding demanded capacities of different resource types are given in $\mathcal{C}^k = \{\boldsymbol{c}_1^k, \boldsymbol{c}_2^k, \ldots, \boldsymbol{c}_{M_k}^k\}$, where $\boldsymbol{c}_m^k = \big(c_m^k(1), c_m^k(2), \ldots, c_m^k(S)\big)$. Each pair of the aforementioned virtual nodes need to be connected if a virtual link between them exists in the set $\mathcal{E}^k = \{e_1^k, e_2^k, \ldots, e_{I_k}^k\}$, and such connectivity should guarantee a minimum data transfer rate specified in $\mathcal{B}^k = \{b_1^k, b_2^k, \ldots, b_{I_k}^k\}$.

The last element, i.e. $\mathcal{L}^k = \{\ell_1^k, \ell_2^k, \ldots, \ell_{M_k}^k\}$, determines the substrate nodes at which each virtual node can be located. We consider the general case, where $\ell_m^k \subseteq \bar{\mathcal{V}}$ shows that the m-th virtual node of request k can be mapped to a subset of substrate nodes. As a special case, $\ell_m^k = \bar{\mathcal{V}}$ implies that such a virtual node does not have any preference for the physical node to be mapped to. For future reference, we also define $M \triangleq \sum_{k=1}^{K} M_k$ and $I \triangleq \sum_{k=1}^{K} I_k$, respectively, as the total number of virtual nodes and virtual links of all requests to be served throughout the network.

Every virtual node belonging to a request's graph can be emblematic of a physical node, a VNF, or a *dummy* node. A dummy node is the one which is not included in the original SFC, is then added in chain re-composition stage, and is neither a VNF nor a physical node (e.g. the ingress/egress nodes where data flow enters/departs the NFVI). As an illustrative example, let us explain the idea by focusing on the three instances of such VNFRs in Fig. 2.

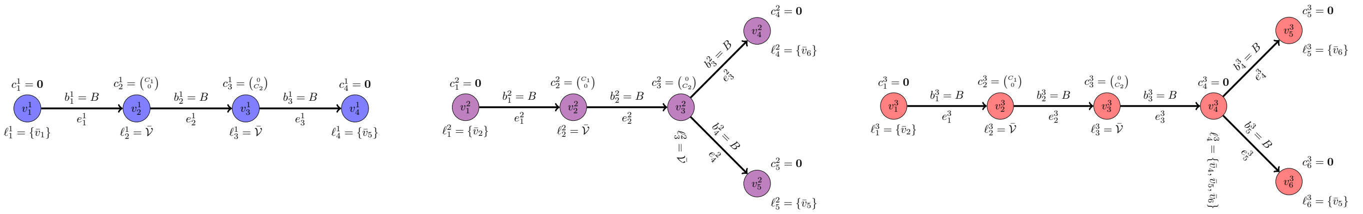
The nodes v_1^1 and v_4^1 in Fig. 2a are two physical nodes, respectively, the source and destination of the first VNFR, whose embedding locations are known and capacities are set to zero; v_2^1 and v_3^1 also represent two VNFs of different types. VNF nodes normally do not have to be embedded in a specific substrate node, so their candidate locations of mapping is the set of entire nodes in NFVI, and their order and required capacities determine their embedding.

In Fig. 2b, in addition to source v_1^2 and VNFs v_2^2 and v_3^2, two destination nodes, namely v_4^2 and v_5^2, are included. Throughout chain re-composition process and especially in the cases with highly correlated SFCs, this structure with multiple sources/destinations is of great benefit. For instance, one may interpret R_2 as a re-composed version of two SFCs of type R_1 with fully correlated data flow, with the same source and different destinations ($\bar{v}_5$ and $\bar{v}_6$). In this case, the resource consumption will be reduced by nearly 50%.

Another case has been illustrated in Fig. 2c. Excluding v_4^3, R_3 is exactly the same as R_2 in terms of its SFC, with source v_1^3, VNFs v_2^3 and v_3^3, and destinations v_5^3 and v_6^3. Thus, the last node to consider, i.e. v_4^3, is a dummy node, an enabler of chain re-composition with zero capacity demanded; its candidate locations, however, implies that it needs to be embedded in the neighborhood of destination nodes. Adding such a node to the SFC helps in postponing the task of splitting data flow, thereby reducing the amount of bandwidth consumed by the request in the substrate network.

In overall, the key observation is that the proposed approach provides a general framework to model VNFRs under a comprehensive range of demand types. Some distinguishing features of such a model are summarized as follows:

- Within such model, no restrictions on topology of re-



(a) Directed graph $\mathcal{G}^1$ representing R_1, where the data flow has to pass through two VNFs of different types.

(b) Directed graph $\mathcal{G}^2$ representing R_2, where the same data flow, after being manipulated by certain VNFs, needs to be broadcast to two destination nodes.

(c) Directed graph $\mathcal{G}^3$ representing R_3, where the to-be-broadcast manipulated data flow has to remain unsplit until having got close to destination zone.

Fig. 2: An example of the requests' set including three VNFRs.

quests' graphs, i.e. $\mathcal{G}^k$'s, are assumed. For instance, the requests with multiple inputs and/or multiple outputs can be shown through a feasible representation as a single graph. As we will show in the following, in some scenarios, such representation can also lead to performance improvement in the final solution obtained.

- Defining vectorized capacities of substrate nodes, i.e. $\bar{\boldsymbol{c}}_n$'s, and virtual nodes, i.e. $\boldsymbol{c}_m^k$'s, allows us to distinguish different resource types, while modeling nodes homogeneously at physical and virtual levels.
- Modeling the candidate locations of each virtual node through ℓ_m^k naturally leads to a more general model where some spatial considerations which influence the mapping of a virtual node can be taken into account. As an example, there might be some nodes, in the substrate network, which are preferable for a certain request in terms of such attributes as security, reliability, or distance.
- The introduction of dummy nodes also plays an important role in enriching the chain re-composition process through exploiting the inherent potential of scenarios such as multiple input/output services, network caching, and correlated SFCs.

C. NFV-MANO

The NFV-MANO needs to choose some of the VNFRs in $\mathcal{R}$ to be served at the current time-slot such that not only are the permitted requests' requirements respected, but the total revenue is maximized as well (*AC* problem). NFV-MANO calculates its revenue with respect to the VNFRs' demanded resources. Subsequently, the orchestrator have to decide how to map the admitted requests to the network's resources (*VNF-FGE* problem). Therefore, in the scope of this paper, there are two tasks in hand to be handled by NFV-MANO, namely AC and VNF-FGE, referred to as *AC/FGE*.

IV. Problem Formulation

In this section, we first specify the variables involved in the problem. These variables will be used to express the constraints and objective of the optimization problem. We also categorize the optimization constraints, where each category represents one of NFV resource allocation stages, namely virtual node mapping, virtual link mapping, joint virtual node/link mapping, and admission control. The first three parts are related to the problem of VNF-FGE. The last part, i.e. AC, can be formulated based on the parameters of VNF-FGE problem. Consequently, we derive the joint optimization problem of AC/FGE.

A. Virtual Node Mapping

We proceed to define a variable which represents how the embedding process of the virtual nodes is done. Consider $\boldsymbol{X}^k = \left[x_{m,n}^k\right] \in \{0,1\}^{M_k \times N}$ as the mapping matrix of virtual nodes of VNFR k, defined as follows:

$$\boldsymbol{X}^k = \begin{array}{cc} \begin{array}{cccc} \bar{v}_1 & \bar{v}_2 & & \bar{v}_N \end{array} & \\ \left[\begin{array}{cccc} x_{1,1}^k & x_{1,2}^k & \dots & x_{1,N}^k \\ x_{2,1}^k & x_{2,2}^k & \dots & x_{2,N}^k \\ \vdots & \vdots & \ddots & \vdots \\ x_{M_k,1}^k & x_{M_k,2}^k & \dots & x_{M_k,N}^k \end{array}\right] & \begin{array}{c} v_1^k \\ v_2^k \\ \\ v_{M_k}^k \end{array} \end{array}, \quad \forall k,$$

where $x_{m,n}^k = 1$ if and only if the m-th virtual node of request k is mapped to substrate node n, and $x_{m,n}^k = 0$ otherwise. Considering the fact that each virtual node can be embedded in not more than one substrate node, it is clear that each row of the above matrix includes at most one non-zero entry.

It is worth mentioning that if VNF splitting is permitted in resource allocation process, then $0 \leqslant x_{m,n}^k \leqslant 1$ represents the percentage of the m-th virtual node corresponding to request k that is served by substrate node n. When mapping variables of the virtual nodes take any value in the range of $[0,1]$, the data to-be-processed enjoys a fine level of granularity such that a flow of data can be divided into a number of sub-flows, each of which would be manipulated by a different VNF instance.

We also define the entire virtual node mapping matrix as

$$\boldsymbol{X} = \left[\begin{array}{c} \boldsymbol{X}^1 \\ \hline \vdots \\ \hline \boldsymbol{X}^K \end{array}\right].$$

Thus, the constraints imposed by virtual node mapping are as follows:

$$\begin{aligned} & \boldsymbol{X} \in \{0,1\}^{M \times N}, && (1) \\ \text{C1:}\quad & x_{m,n}^k = 0, \quad \forall k, m, n : \bar{v}_n \notin \ell_m^k, \\ \text{C2:}\quad & \sum_{k=1}^{K} \sum_{m=1}^{M_k} \boldsymbol{c}_m^k(s) x_{m,n}^k \leqslant \bar{\boldsymbol{c}}_n(s), \quad \forall n, s, \end{aligned}$$

where (1) has to be substituted with (2) if VNF splitting is permissible:

$$\boldsymbol{X} \in [0,1]^{M \times N}. \tag{2}$$

The constraint C1 prevents the virtual nodes from being mapped to the substrate nodes which are not included in the permitted locations, and constraint C2 considers substrate nodes' capacities of different resource types.

B. Virtual Link Mapping

To determine the routing of VNFRs' flows mapped to the NFVI, we need to consider the restrictions imposed by the limited capacity of substrate network's links on how nodes can communicate with each other and consequently on the allocation of bandwidth resources to requested virtual links.

First, assume that J possible simple paths can be found in the substrate network, i.e. $\bar{\mathcal{G}}$. These paths are listed in a set $\bar{\mathcal{P}} = \{\bar{P}_1, \bar{P}_2, \ldots, \bar{P}_J\}$. Each path $\bar{P}_j = (\bar{v}_{n_{j,1}}, \bar{v}_{n_{j,2}}, \ldots, \bar{v}_{n_{j,Q_j}})$ of length $Q_j - 1$ is an ordered Q_j-tuple of substrate nodes, where $n_{j,q} \in \{1, 2, \ldots, N\}, \forall q \in \{1, 2, , \ldots, Q_j\}$. For a feasible path $\bar{P}_j$, a substrate link should exist between each pair of consecutive nodes in $\bar{P}_j$, or $(\bar{v}_{n_{j,q}}, \bar{v}_{n_{j,q+1}}) \in \bar{\mathcal{E}}, \forall q \in \{1, 2, \ldots, Q_j - 1\}$, and $\bar{P}_j$ should not traverse a substrate node more than once, i.e. $\bar{v}_{n_{j,q}} \neq \bar{v}_{n_{j,q'}}$ if $q \neq q', \forall q, q' \in \{1, 2, \ldots, Q_j - 1\}$. Such node-based representation helps determine the sources and destinations of substrate paths.

In order to specify the substrate links included in a path, a $(Q_j - 1)$-edge-long simple path can be expressed as a sequence of $Q_j - 1$ substrate edges, $\bar{P}_j = (\bar{e}_{f_{j,1}}, \bar{e}_{f_{j,2}}, \ldots, \bar{e}_{f_{j,Q_j-1}})$, where $f_{j,q} \in \{1, 2, \ldots, F\}, \forall q \in \{1, 2, , \ldots, Q_j - 1\}$. In addition, each two consecutive links in the path have exactly one node in common, i.e. $\forall q \neq q', q, q' \in \{1, 2, \ldots, Q_j - 1\}, |\bar{e}_{f_{j,q}} \cap \bar{e}_{f_{j,q'}}| = 1$ if $|q - q'| = 1$ and $|\bar{e}_{f_{j,q}} \cap \bar{e}_{f_{j,q'}}| = 0$ otherwise.[1]

In order to represent the mapping of the k-th VNFR's virtual links to substrate paths, the matrix $\boldsymbol{Y}^k$ composed of $I_k J$ binary variables is defined as follows:

$$\boldsymbol{Y}^k = \begin{array}{cc} \begin{array}{cccc} \bar{P}_1 & \bar{P}_2 & & \bar{P}_J \end{array} & \\ \left[\begin{array}{cccc} y^k_{1,1} & y^k_{1,2} & \cdots & y^k_{1,J} \\ y^k_{2,1} & y^k_{2,2} & \cdots & y^k_{2,J} \\ \vdots & \vdots & \ddots & \vdots \\ y^k_{I_k,1} & y^k_{I_k,2} & \cdots & y^k_{I_k,J} \end{array}\right] & \begin{array}{c} e^k_1 \\ e^k_2 \\ \\ e^k_{I_k} \end{array} \end{array}, \quad \forall k,$$

where $y^k_{i,j}$ is the variable showing whether or not the i-th virtual link of request k is mapped to path $\bar{P}_j$. More precisely, $y^k_{i,j}$ is set to 1 if the mapping is carried out and 0 otherwise. The entire virtual link mapping matrix is also formed as:

$$\boldsymbol{Y} = \left[\begin{array}{c} \boldsymbol{Y}^1 \\ \hline \vdots \\ \hline \boldsymbol{Y}^K \end{array}\right].$$

It should be noted that, in the case where data flow is allowed to be split while passing through a substrate node, each virtual link can be mapped to more than one physical path, as indicated by the fact that $\boldsymbol{Y} \in [0,1]^{I \times J}$. Consequently, the constraints can be written as:

$$\boldsymbol{Y} \in \{0,1\}^{I \times J}, \tag{3}$$

$$\text{C3:} \quad \sum_{k=1}^{K} \sum_{i=1}^{I_k} \sum_{\substack{j=1 \\ \exists q: f_{j,q} = f}}^{J} b^k_i y^k_{i,j} \leqslant \bar{b}_f, \quad \forall f,$$

where we replace (3) with (4) when multi-path routing is allowed:

$$\boldsymbol{Y} \in [0,1]^{I \times J}, \tag{4}$$

and constraint C3 ensures that the amount of data to be communicated through a substrate link does not violate its bandwidth capacity.

C. Joint Virtual Node/Link Mapping

Considering the interrelation of virtual node and link mapping variables, and to ensure the balance between amount of the input data of each substrate node and the amount of data it processes, we define matrix $\hat{\boldsymbol{Y}}$ with entries $\hat{y}^k_{i,n}$ as:

$$\hat{\boldsymbol{Y}}^k = \begin{array}{cc} \begin{array}{cccc} \hat{P}_1 & \hat{P}_2 & & \hat{P}_N \end{array} & \\ \left[\begin{array}{cccc} \hat{y}^k_{1,1} & \hat{y}^k_{1,2} & \cdots & \hat{y}^k_{1,N} \\ \hat{y}^k_{2,1} & \hat{y}^k_{2,2} & \cdots & \hat{y}^k_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{y}^k_{I_k,1} & \hat{y}^k_{I_k,2} & \cdots & \hat{y}^k_{I_k,N} \end{array}\right] & \begin{array}{c} e^k_1 \\ e^k_2 \\ \\ e^k_{I_k} \end{array} \end{array}, \quad \forall k,$$

where $\hat{P}_n, n \in \{1, 2, \ldots, N\}$ represents the internal interface within a substrate node $\bar{v}_n$, and

$$\hat{\boldsymbol{Y}} = \left[\begin{array}{c} \hat{\boldsymbol{Y}}^1 \\ \hline \vdots \\ \hline \hat{\boldsymbol{Y}}^K \end{array}\right].$$

In Fig. 3, consider v and v' as internal processors (e.g. virtual machines and containers [35]) performing a share of computing tasks related to v^k_m and $v^k_{m'}$, respectively. Also, let $\hat{P}_n$ be the internal path (e.g. virtual switches [36]) facilitating a medium of communication between these nodes inside the substrate node $\bar{v}_n$. $\{\bar{P}_j\}$ and $\{\bar{P}_{j'}\}$ are the sets of paths which bear outward and inward data flows of e^k_i from/to node $\bar{v}_n$, respectively (note that the paths traversing $\bar{v}_n$ in the middle of their routes are not included in either of these sets).

As shown in Fig. 3, for each virtual link $e^k_i = (v^k_m, v^k_{m'})$ and each substrate node $\bar{v}_n$, the following node and link mapping variables are related to both e^k_i and $\bar{v}_n$: 1) $x^k_{m,n}$ and $x^k_{m',n}$ respectively show the percentage of v^k_m and $v^k_{m'}$ mapped to substrate node $\bar{v}_n$. 2) $y^k_{i,j}$'s (where $\bar{v}_n$ is an end-point of path $\bar{P}_j$) are indicators of the portion of e^k_i served by the paths which are terminated in $\bar{v}_n$. 3) $\hat{y}^k_{i,n}$ determines how much data processed in $\bar{v}_n$ is transferred internally when $\bar{v}_n$ is allocated to both virtual nodes of a virtual link, i.e. $x^k_{m,n} x^k_{m',n} \neq 0$.

In order to reveal the relation between the aforementioned variables, we use the fact that, for each internal processor,

[1] In order to reduce the complexity of finding all possible paths in the network as well as dimension of optimization problem's variable, one may either consider only the shortest path (or alternatively the k shortest paths) between each pair of substrate nodes (using methods such as Dijkstra's and Yen's algorithms [33], [34]) or include only the paths with delays (or hop counts) not more than a certain amount.

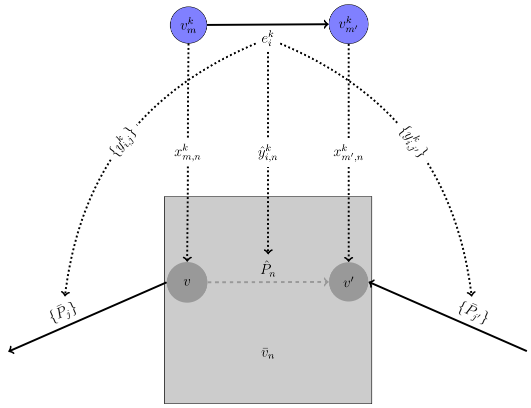


Fig. 3: Illustration of the variables involved in mapping of a virtual link and its end-nodes to a certain substrate node and its corresponding paths (internal processing and communication inside the substrate node are depicted as v, v', and $\hat{P}_n$).

a balance between the amounts of its processing and communication should be maintained. This, according to Fig. 3, results in the equations $x_{m,n}^k = \sum_j y_{i,j}^k + \hat{y}_{i,n}^k$ and $x_{m',n}^k = \sum_{j'} y_{i,j}^k + \hat{y}_{i,n}^k$, for v and v', respectively. Added together, these equations form the constraints related to each virtual link and substrate node.

Subsequently, the additional constraints resulting from mutual dependence of two problems, namely virtual node mapping and virtual link mapping, are:

$$\hat{\boldsymbol{Y}} \in [0,1]^{I \times N}, \tag{5}$$

$$\text{C4:} \quad \sum_{\substack{m=1 \\ v_m^k \in e_i^k}}^{M_k} x_{m,n}^k = \sum_{\substack{j=1 \\ n \in \{n_{j,1}, n_{j,Q_j}\}}}^{J} y_{i,j}^k + 2\hat{y}_{i,n}^k, \quad \forall k, i, n,$$

$$\text{C5:} \quad \hat{y}_{i,n}^k \leqslant \min_{v_m^k \in e_i^k} x_{m,n}^k, \quad \forall k, i, n,$$

where constraint C4 guarantees the balance between the amount of processing and communication which are requested by e_i^k and performed in substrate node $\bar{v}_n$. Also, since the amount of data communicated through the internal interface cannot exceed the amount of data corresponding to either v_m^k or $v_{m'}^k$ which is internally processed at node $\bar{v}_n$, C5 should hold true.

D. Admission Control

In order to relate the problems of AC and VNF-FGE, we denote the fraction of each request R_k which is mapped to the substrate network resources by α_k, calculated as follows:

$$\alpha_k \triangleq \sum_{n=1}^{N} x_{1,n}^k, \quad \forall k. \tag{6}$$

If $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_K)$ denotes the *admission vector*, the feasible solutions of AC problem can be expressed as $\boldsymbol{\alpha} \in \{0,1\}^K$, where it should not only meet the requirements of the accepted requests (the ones corresponding to 1 entries of $\boldsymbol{\alpha}$), but also should not violate the constraints on the capacity of substrate network resources when allocated to the corresponding admitted requests.

The AC problem, which, in addition to VNF-FGE problem, forms the entire AC/FGE, is subject to the following conditions:

$$\text{C6:} \quad \sum_{n=1}^{N} x_{m,n}^k \leqslant 1, \quad \forall k, m,$$

$$\text{C7:} \quad \sum_{n=1}^{N} x_{m,n}^k = \sum_{j=1}^{J} y_{i,j}^k + \sum_{n=1}^{N} \hat{y}_{i,n}^k, \quad \forall k, m, i,$$

$$\text{C8:} \quad \boldsymbol{\alpha} \in \{0,1\}^K.$$

The requirement of serving each request at most once is specified in C6. C7 arises as a VNFR (including all of its virtual nodes and links) should be either completely served or not served at all. Consequently, node embedding process cannot be accomplished when, for a specific request, a portion of virtual nodes are mapped properly, while the rest remain unserved. Finally, C8 illustrates the intrinsic property of admission vector, having to be a set of binary values.

According to (6), one may notice that C8 is a constraint on mapping variables, since admission variables are completely expressed based on them. In an uncoordinated manner, however, the admission vector is initialized employing an external AC process, with respect to C8 and independently of mapping variables. Then, the mapping variables will be subject to the decision made about admission variables beforehand.

In the AC/FGE problem, we consider the objective of revenue maximization, where revenue corresponds to the benefit of serving VNFRs to be served. The revenue of each request can be defined similar to the one in [20] as follows:

$$r_k \triangleq \sum_{m=1}^{M_k} \sum_{s=1}^{S} \eta_s c_m^k(s) + \sum_{i=1}^{I_k} b_i^k, \quad \forall k,$$

with $\eta_s, s \in \{1, 2, \ldots, S\}$ being the ratio of the revenue of allocating a unit of s-type resource to the revenue per bandwidth unit. Consequently, $\boldsymbol{r} = (r_1, r_2, \ldots, r_K)$ can be considered to be the revenue vector.

Let $\boldsymbol{z}$ be the optimization variable, a vector of size $D \triangleq MN + IJ + IN$ and containing all entries of $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\hat{\boldsymbol{Y}}$. Therefore, the AC/FGE optimization problem in case of unsplittable VNF and non-permissible multi-path routing, referred to as *hard AC/FGE*, can be expressed as the following ILP:

$$\max_{\boldsymbol{z} \in \{0,1\}^D} \boldsymbol{r}^T \boldsymbol{\alpha} \tag{7}$$
$$\text{s.t.} \quad \text{C1–C7},$$

where, in case of hard AC/FGE, (5) converts to the binary condition, and C8 automatically holds true.

On the other hand, for *soft AC/FGE*, where VNF splitting and multi-path routing are allowed, we need to deal with the following MILP problem:

$$\max_{\boldsymbol{z} \in [0,1]^D} \boldsymbol{r}^T \boldsymbol{\alpha} \tag{8}$$
$$\text{s.t.} \quad \text{C1–C8}.$$

In order to perceive how the mapping variables are related to the VNF-FGE problem, a possible solution of soft AC/FGE in (8) for the set of VNFRs specified in Fig. 2 over the NFVI of Fig. 1 is presented in Fig. 4. In this example, R_1 and R_3 are admitted, while R_2 is rejected, due to limited capacities of NFVI resources. Non-zero mapping variables of virtual nodes and of virtual links are determined in Figs. 4a and 4b, respectively. As can be seen, R_1 is embedded in NFVI without either VNF or path splitting. However, in the process of mapping R_3, VNF splitting (v_2^3 mapped to $\bar{v}_3$ and $\bar{v}_4$) and multi-path routing (e_1^3 mapped to $\bar{P}_6$ and $\bar{P}_7$, and e_2^3 mapped to $\bar{P}_{10}$ and $\hat{P}_4$) are needed. In addition, when two connected virtual nodes are mapped to the same substrate node, the virtual link between the two virtual nodes is mapped to the internal path of the substrate node (e_2^1, e_2^3, and e_3^3 mapped to $\hat{P}_4$, and e_3^1 mapped to $\hat{P}_5$).

V. Hard AC/FGE Proposed Solution

In this section, we propose a method to find the sub-optimum solution of hard AC/FGE formulated in (7), where AC/FGE problem is subject to no VNF splitting or multi-path routing. To tackle such NP-hard problem, we adopt an SCA approach, which locally optimizes non-convex problems by leveraging convex optimization. To apply the SCA method, first, we eliminate the equality constraints, leading to the reduction of optimization variable dimension. Next, a reformulation of the original ILP is introduced, which replaces the linear objective function by ℓ_2-norm, while relaxing the binary constraint. Then, we relax a constraint of the resulting optimization problem, and subsequently a quantization process has to be done to ensure the feasibility of the solution. In the relaxed problem, the ℓ_2-norm objective function helps approach the binary solution. Finally, we use an SCA method to deal with the norm maximization problem.

A. Optimization Variable Dimension Reduction

Let us divide $\boldsymbol{z}$ into two blocks as $\boldsymbol{z} = (\tilde{\boldsymbol{z}}, \tilde{\boldsymbol{z}}_e)$, where $\boldsymbol{z} = \boldsymbol{C}\tilde{\boldsymbol{z}}$ is equivalent to the set of linear equality constraints $(\mathrm{C1}, \mathrm{C4}, \mathrm{C7})$, for a matrix $\boldsymbol{C} \in \mathbb{R}^{D\times\tilde{D}}$ if the size of $\tilde{\boldsymbol{z}}$ equals $\tilde{D} \leqslant D$. In other words, based on $D-\tilde{D}$ independent equality constraints of (7), we reduce the dimension of the optimization problem by $D-\tilde{D}$, eliminating the elements specified in $\tilde{\boldsymbol{z}}_e$. It also should be noted that we assume that among the elements $x_{1,n}^k, \forall k \in \{1, 2, \ldots, K\}, \forall n \in \{1, 2, \ldots, N\}$, the ones which are non-zero according to C1 have to be included in $\tilde{\boldsymbol{z}}$.

Therefore, adopting $\boldsymbol{z} = \boldsymbol{C}\tilde{\boldsymbol{z}}$, we can replace (7) with the following optimization problem, with only linear inequality constraints:

$$\max_{\tilde{\boldsymbol{z}} \in \{0,1\}^{\tilde{D}}} \boldsymbol{r}^T \boldsymbol{\alpha} = \sum_{k=1}^{K} \sum_{\substack{n=1 \\ \bar{v}_n \in \ell_m^k}}^{N} r_k x_{1,n}^k \tag{9}$$
$$\text{s.t.} \quad \mathrm{C2}, \mathrm{C3}, \mathrm{C5}, \mathrm{C6},$$

where the constraints are expressed based on $\tilde{\boldsymbol{z}}$.

B. Reformulation

In the next step, we mention the following theorem, which introduces a reformulation to relax the binary constraint on the optimization problem.

Theorem 1. *For any vector $\boldsymbol{r} = (r_1, r_2, \ldots, r_{\hat{D}}) \in (0, \infty)^{\hat{D}}$, consider the following optimization problem:*

$$\mathcal{P}_1 : \max_{\boldsymbol{z}=(\hat{\boldsymbol{z}},\check{\boldsymbol{z}})} f_1(\boldsymbol{z}) \triangleq \underbrace{\sum_{d=1}^{\hat{D}} r_d \hat{z}_d}_{g_1(\hat{\boldsymbol{z}})}$$
$$\text{s.t.} \quad \boldsymbol{z} \in \Omega, \tag{10}$$
$$\boldsymbol{z} \in \{0,1\}^D, \tag{11}$$

where Ω is a convex set, the feasible region of $\mathcal{P}_1$ is non-empty, $\hat{\boldsymbol{z}} = (\hat{z}_1, \hat{z}_2, \ldots, \hat{z}_{\hat{D}})$, $\check{\boldsymbol{z}} = (\check{z}_1, \check{z}_2, \ldots, \check{z}_{\check{D}})$, and $D = \hat{D} + \check{D}$. Also suppose that R^* is the optimal value for $\mathcal{P}_1$. Note that $\hat{\boldsymbol{z}}$ is composed of the variables contributing to the objective function, while the remainder of $\boldsymbol{z}$ is included in $\check{\boldsymbol{z}}$.

We define another optimization problem by altering the objective function, for any $\delta > 0$, and relaxing the binary constraint as below:

$$\mathcal{P}_2 : \max_{\boldsymbol{z}=(\hat{\boldsymbol{z}},\check{\boldsymbol{z}})} f_2(\boldsymbol{z}) \triangleq \underbrace{\sum_{d=1}^{\hat{D}} r_d \hat{z}_d^2}_{g_2(\hat{z})} + \underbrace{\delta \sum_{d=1}^{\check{D}} (\check{z}_d - \frac{1}{2})^2}_{h_2(\check{z})}$$
$$\text{s.t.} \quad \boldsymbol{z} \in \Omega, \tag{12}$$
$$\boldsymbol{z} \in [0,1]^D, \tag{13}$$
$$\sum_{d=1}^{\hat{D}} r_d \hat{z}_d \leqslant R^*. \tag{14}$$

It can be shown that $\boldsymbol{z}^*$ is an optimum solution for $\mathcal{P}_1$, if and only if $\boldsymbol{z}^*$ is an optimum solution for $\mathcal{P}_2$.

Proof. See Appendix A. □

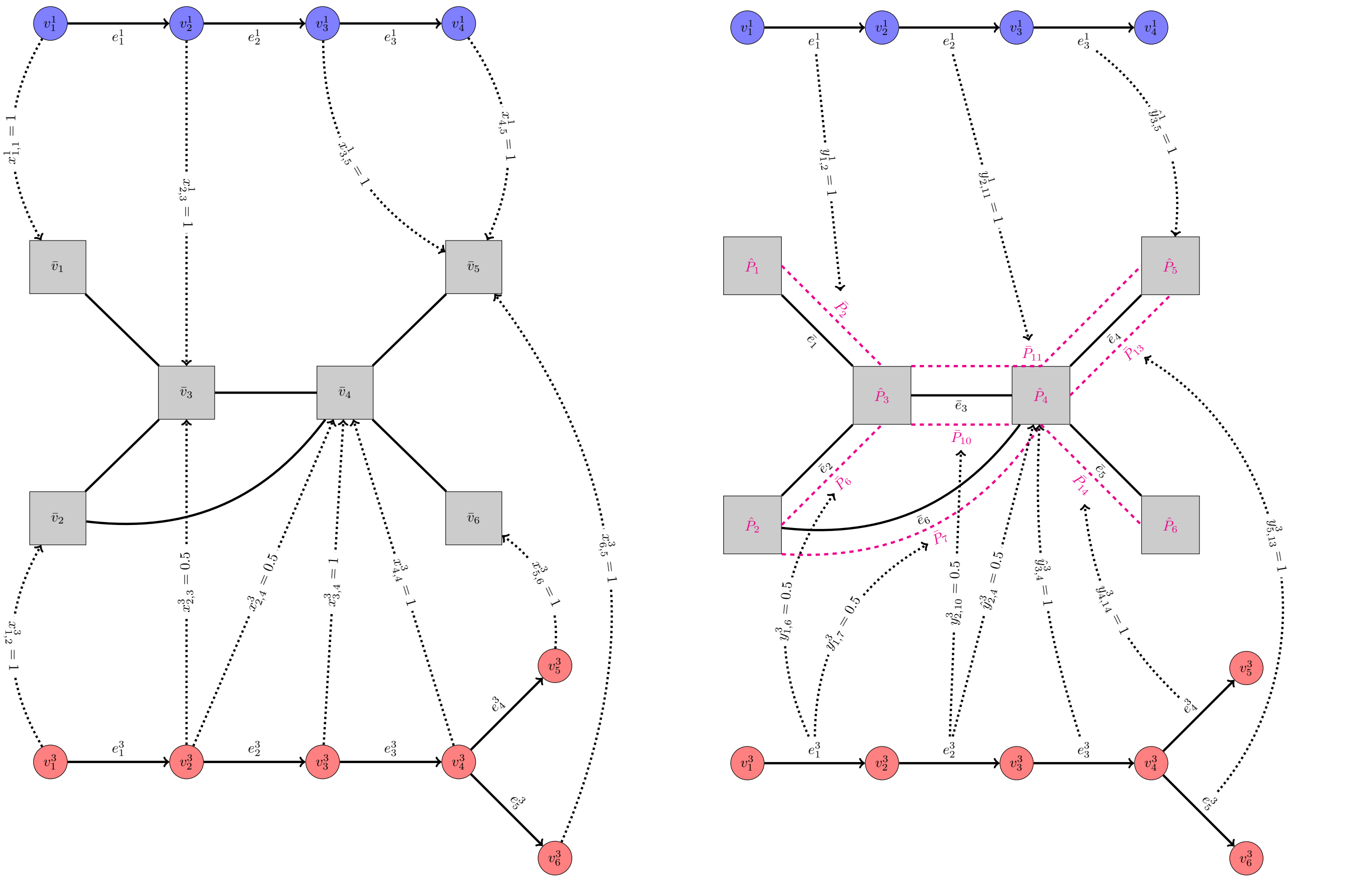
We split the optimization variable of (9) into two blocks as $\tilde{\boldsymbol{z}} = (\hat{\tilde{\boldsymbol{z}}}, \check{\tilde{\boldsymbol{z}}})$, where the elements contributing to the objective function are included in $\hat{\tilde{\boldsymbol{z}}}$. According to Theorem 1, we can look for a solution to the following optimization problem, as a substitute for (9):

$$\max_{\tilde{\boldsymbol{z}} \in [0,1]^{\tilde{D}}} \sum_d r_{k_d} \hat{\tilde{z}}_d^2 + \delta \sum_d (\check{\tilde{z}}_d - \frac{1}{2})^2 \tag{15}$$
$$\text{s.t.} \quad \mathrm{C2}, \mathrm{C3}, \mathrm{C5}, \mathrm{C6},$$
$$\sum_d r_{k_d} \hat{\tilde{z}}_d \leqslant R^*, \tag{16}$$

where $k_d = k$ if $\hat{\tilde{z}}_d \equiv x_{1,n}^k$.

C. Relaxation and Quantization

Relaxing all binary constraints, such a modification leads to an optimization problem where all constraints are affine inequalities. However, there are still two issues with the optimization problem in (15). First, the objective is to maximize a convex function, which is of NP-hard complexity [37], and secondly, R^* is not known a-priori. To address the first issue, we employ the SCA method proposed in [38] (explained in Appendix B). For the latter issue, we eliminate constraint (16), therefore, broadening the feasible region. Subsequently, the final solution requires some modification to conform with



(a) Virtual node mapping (with non-zero entries of $\boldsymbol{X}$).

(b) Virtual link mapping (with non-zero entries of $[\boldsymbol{Y}|\hat{\boldsymbol{Y}}]$).

Fig. 4: A soft AC/FGE optimum solution based on (8) with respect to the NFVI and the request set in Figs. 1 and 2, respectively (where, in AC process, R_1 and R_3 are admitted, and R_2 is rejected).

the binary constraint on $\tilde{z}$. To this end, we need to perform a quantization on the solution of the altered optimization problem. It should be noted that the quantization process approximates the elements with respect to a threshold level of $1-\epsilon$, where ϵ represents the distance within which the output of the norm maximization method in Algorithm A is located from the sub-optimum solution. Consequently, the following optimization problem, to find the solution of (15), is obtained:

$$\begin{aligned}
\max_{\tilde{\boldsymbol{z}}\in[0,1]^{\tilde{D}}} \quad & \sum_d r_{k_d}\hat{\tilde{z}}_d^2 + \delta \sum_d (\check{\tilde{z}}_d - \frac{1}{2})^2 && (17)\\
\text{s.t.} \quad & \tilde{\text{C}}2\text{: } \sum_{k=1}^{K}\sum_{m=1}^{M_k} c_m^k(s) x_{m,n}^k \leqslant (1-\epsilon)\bar{c}_n(s), \quad \forall n, s,\\
& \tilde{\text{C}}3\text{: } \sum_{k=1}^{K}\sum_{i=1}^{I_k}\sum_{\substack{j=1\\ \exists q: f_{j,q}=f}}^{J} b_i^k y_{i,j}^k \leqslant (1-\epsilon)\bar{b}_f, \quad \forall f,\\
& \text{C5, C6.}
\end{aligned}$$

In (17), we replaced C2 and C3 by $\tilde{\text{C}}2$ and $\tilde{\text{C}}3$ respectively to ensure the feasibility after quantizing the entries of the solution (the entries less than $1-\epsilon$ get rounded down to 0, and rounded up to 1 otherwise).

D. Norm Maximization Standard Form

Since the constraints ($\tilde{\boldsymbol{z}} \in [0,1]^{\tilde{D}}, \tilde{\text{C}}2, \tilde{\text{C}}3, \text{C}5, \text{C}6$) are linear inequalities, their feasible region can be represented by a matrix $\boldsymbol{U} \in \mathbb{R}^{\tilde{E}\times\tilde{D}}$ and a vector $\boldsymbol{v} \in \mathbb{R}^{\tilde{E}}$, where $\tilde{E}$ is the number of all inequality constraints. Thus, the vectorized version of (17) can be formulated as:

$$\begin{aligned}
\max_{\tilde{\boldsymbol{z}}\in\mathbb{R}^{\tilde{D}}} \quad & \|\boldsymbol{R}(\tilde{\boldsymbol{z}}-\boldsymbol{s})\|_2 && (18)\\
\text{s.t.} \quad & \boldsymbol{U}\tilde{\boldsymbol{z}} \leqslant \boldsymbol{v},
\end{aligned}$$

where $\boldsymbol{R} = \left[r_{d,d'}\right] \in \mathbb{R}^{\tilde{D}\times\tilde{D}}$ is an invertible diagonal matrix with:

$$r_{d,d'} = \begin{cases} \sqrt{r_{k_{\hat{d}}}}, & d = d', \tilde{z}_d \equiv \hat{\tilde{z}}_{\hat{d}},\\ \sqrt{\delta}, & d = d', \tilde{z}_d \equiv \check{\tilde{z}}_{\check{d}},\\ 0, & d \neq d', \end{cases} \quad \forall d, d', \tag{19}$$

and $\boldsymbol{s} = (s_1, s_2, \ldots, s_{\tilde{D}}) \in \mathbb{R}^{\tilde{D}}$ is a vector with entries:

$$s_d = \begin{cases} 0, & \tilde{z}_d \equiv \hat{\tilde{z}}_{\hat{d}},\\ \frac{1}{2}, & \tilde{z}_d \equiv \check{\tilde{z}}_{\check{d}}, \end{cases} \quad \forall d. \tag{20}$$

By defining a new variable $\tilde{\tilde{\boldsymbol{z}}} \triangleq \boldsymbol{R}(\tilde{\boldsymbol{z}}-\boldsymbol{s})$, the optimization problem in (18) can be expressed in terms of $\tilde{\tilde{\boldsymbol{z}}}$ in the following

Algorithm 1: Hard AC/FGE Proposed Method

```
Input: $\bar{\mathcal{G}}$, $\mathcal{R}$, $\epsilon$
Output: $\boldsymbol{z}^*$
Initialization:
Determine the matrix $\boldsymbol{C}$ such that $\boldsymbol{z} = \boldsymbol{C}\tilde{\boldsymbol{z}}$ is equivalent to the set of linear equality constraints $(\mathrm{C1}, \mathrm{C4}, \mathrm{C7})$.
Determine matrix $\boldsymbol{U}$ and vector $\boldsymbol{v}$ such that $\{\tilde{\boldsymbol{z}} | \boldsymbol{U}\tilde{\boldsymbol{z}} \leqslant \boldsymbol{v}\}$ is equivalent to the feasible region specified by constraints $(\tilde{\boldsymbol{z}} \in [0,1]^{\tilde{D}}, \tilde{\mathrm{C}}2, \tilde{\mathrm{C}}3, \mathrm{C5}, \mathrm{C6})$.
Construct the matrix $\boldsymbol{R}$ and vector $\boldsymbol{s}$ according to (19) and (20) respectively.
$\tilde{\boldsymbol{U}} \leftarrow \boldsymbol{U}\boldsymbol{R}^{-1}$
$\tilde{\boldsymbol{v}} \leftarrow \boldsymbol{v} - \boldsymbol{U}\boldsymbol{s}$
Step 1: Solve (21), by employing an SCA method:
$\tilde{\tilde{\boldsymbol{z}}}^* \leftarrow \textsc{NormMax}(\tilde{\boldsymbol{U}}, \tilde{\boldsymbol{v}}, \epsilon)$ [a]
Step 2: Find the solution of (18), according to the linear transformation $\tilde{\tilde{\boldsymbol{z}}} = \boldsymbol{R}(\tilde{\boldsymbol{z}} - \boldsymbol{s})$ and invertibility of $\boldsymbol{R}$:
$\tilde{\boldsymbol{z}}^* \leftarrow \boldsymbol{R}^{-1}\tilde{\tilde{\boldsymbol{z}}}^* + \boldsymbol{s}$
Step 3: Modify the solution of previous step to comply with the binary constraint of (9). Every entry of $\tilde{\boldsymbol{z}}^*$ which is less than $1 - \epsilon$ is reset to 0, and set to 1 otherwise.
Step 4: Retrieve the AC/FGE solution of (7) from $\tilde{\boldsymbol{z}}^*$, using the equality constraints:
$\boldsymbol{z}^* \leftarrow \boldsymbol{C}\tilde{\boldsymbol{z}}^*$
```

[a] This function is a norm maximizer introduced in Algorithm A

manner:

$$\max_{\tilde{\tilde{\boldsymbol{z}}} \in \mathbb{R}^{\tilde{D}}} \; \|\tilde{\tilde{\boldsymbol{z}}}\|_2 \qquad \text{s.t.} \quad \tilde{\boldsymbol{U}}\tilde{\tilde{\boldsymbol{z}}} \leqslant \tilde{\boldsymbol{v}}, \tag{21}$$

where $\tilde{\boldsymbol{U}} \triangleq \boldsymbol{U}\boldsymbol{R}^{-1}$ and $\tilde{\boldsymbol{v}} \triangleq \boldsymbol{v} - \boldsymbol{U}\boldsymbol{s}$. As it can be verified, the optimization problem in (21) is in the form of standard norm maximization problem in (26) (refer to Appendix B). Therefore, the method presented in Algorithm A can be used to find the sub-optimum solution $\tilde{\tilde{\boldsymbol{z}}}^*$, and subsequently $\tilde{\boldsymbol{z}}^*$ is computed as $\tilde{\boldsymbol{z}}^* = \boldsymbol{R}^{-1}\tilde{\tilde{\boldsymbol{z}}}^* + \boldsymbol{s}$ with entries less than $1 - \epsilon$ rounded down to 0, and rounded up to 1 otherwise, in order to ensure the binary constraint. Finally, the AC/FGE solution $\boldsymbol{z}^*$ is obtained using the equality constraints as $\boldsymbol{z}^* = \boldsymbol{C}\tilde{\boldsymbol{z}}^*$. The entire procedure has been described in Algorithm 1.

Finally, the following shows that the relaxation/quantization procedure in Algorithm 1 maintains the feasibility of the final solution.

Proposition 1. *For any* $0 \leqslant \epsilon < 0.5$, *if* $\tilde{\boldsymbol{z}}$ *is a feasible point of* (17), *then* $\bar{\boldsymbol{z}}$ *is a feasible point of* (9), *where* $\bar{\boldsymbol{z}}$ *is obtained by performing the quantization procedure in Algorithm 1 on* $\tilde{\boldsymbol{z}}$.

Proof. See Appendix C. □

Theorem 2. *The output of Algorithm 1 is a hard AC/FGE feasible solution.*

Proof. Considering Proposition 1, the output of the third step of Algorithm 1 belongs to the feasible region of (9). Therefore, all the inequality constraints of (7) are satisfied. Since the fourth step is to ensure that the equality constraints of (7) are satisfied, the final solution of Algorithm 1 necessarily meets all the constraints of (7), completing the proof. □

VI. Soft AC/FGE Proposed Solution

In this section, we consider soft AC/FGE introduced in (8). By exploiting the linear property of the resulting MILP, we propose a heuristic approach to find the sub-optimum solution of soft AC/FGE.

An approach to achieve a low-complexity method for solving soft AC/FGE is to decouple the process of handling the two problems involved in the procedure, namely AC and VNF-FGE. Since VNF-FGE in soft AC/FGE is a linear programming problem, such a separation helps perform AC in a heuristic manner, and then to find the optimum solution of VNF-FGE with respect to the output of AC stage. To this end, we propose an algorithm which starts with an all-one admission vector and, by gradually lowering the number of admitted requests, terminates at the point where a feasible solution to the corresponding VNF-FGE problem is obtained.

If the admission vector is assumed to be fixed, constraint C8 becomes affine. Considering the fact that all other constraints C1–C7 in the optimization problem of soft AC/FGE are affine, we arrive at a linear programming problem. We set the admission vector to $\mathbf{1}$ to ensure that the requirements of all requests are met. To assure the feasibility of the problem, we define some additional variables $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_N)$ and $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_F)$, which are the scale factors related to the capacity of substrate nodes and links, respectively. By performing the first step of Algorithm 2, arriving at the optimal solution of 1 for a scale factor implies that the capacity of the corresponding node/link is sufficient to handle the admitted requests with respect to $\tilde{\boldsymbol{\alpha}}$. On the other hand, the resources whose scale factors' optimum solutions tend to be greater than 1 are considered not to be adequate to serve the admitted VNFRs. Consequently, the overloaded resources are discovered, and, in the second step, we determine the dominant VNFR occupying the resource corresponding to the largest scale factor. Then, through the AC stage, the aforementioned VNFR is rejected, and VNF-FGE is performed over the remaining requests to check if a feasible AC solution is achieved.

The entire proposed heuristic approach is shown in Algorithm 2. It should be noted that, although the final solution falls short of optimality due to decomposition of AC and VNF-FGE problems, the embedding of the requests' forwarding graphs is still performed by considering virtual node and link mappings together and solving the VNF-FGE optimization in a joint manner.

VII. Evaluation and Discussion

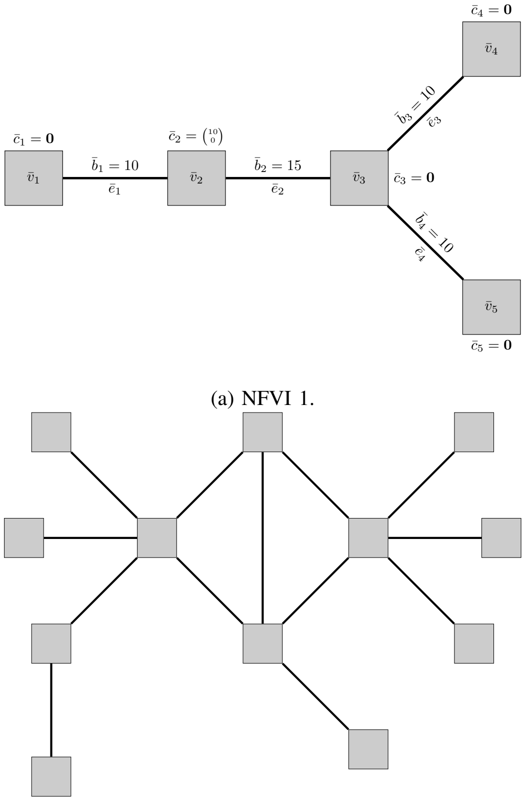
This section evaluates the improvement made by recomposition stage, suggested in Section III, as well as the performance achieved by the proposed methods to solve hard and soft AC/FGE, introduced in Sections V and VI, respectively. We draw comparisons between each of the aforementioned algorithms and a baseline algorithm as well as the optimum solution. The baseline algorithm considered is the heuristic method proposed in [20], which performs AC and VNF-FGE with the aim of revenue maximization. It also handles the batch of arriving VNFRs together at a time, which makes

```
Algorithm 2: Soft AC/FGE Proposed Method
Input: $\bar{\mathcal{G}}$, $\mathcal{R}$
Output: $\boldsymbol{z}^*$
Initialization: $\tilde{\mathcal{K}} \leftarrow \{1, 2, \dots, K\}$, $\tilde{\boldsymbol{\alpha}} \leftarrow \boldsymbol{1}$
Step 1: Solve the following linear programming, with $\boldsymbol{z}^*$, $\boldsymbol{\gamma}^*$, and $\boldsymbol{\lambda}^*$ being the optimum solutions:
    $$\min_{\substack{\boldsymbol{z} \in [0,1]^D \\ \boldsymbol{\gamma} \in [1,\infty)^N \\ \boldsymbol{\lambda} \in [1,\infty)^F}} \sum_{n=1}^{N} \gamma_n + \sum_{f=1}^{F} \lambda_f$$
    $$\text{s.t.} \quad \text{C1, C4–C7},$$
    $$\sum_{k=1}^{K} \sum_{m=1}^{M_k} c_m^k(s) x_{m,n}^k \leqslant \gamma_n \bar{c}_n(s), \quad \forall n, s,$$
    $$\sum_{k=1}^{K} \sum_{i=1}^{I_k} \sum_{\substack{j=1 \\ \exists q: f_{j,q} = f}}^{J} b_i^k y_{i,j}^k \leqslant \lambda_f \bar{b}_f, \quad \forall f,$$
    $$\boldsymbol{\alpha} = \tilde{\boldsymbol{\alpha}}.$$
Step 2:
if $\boldsymbol{\gamma}^* = \boldsymbol{1}$ and $\boldsymbol{\lambda}^* = \boldsymbol{1}$ then
    Terminate the procedure.
end
else
    Let $\bar{v}_{\tilde{n}}$ ($\bar{e}_{\tilde{f}}$) be the corresponding substrate node (link) to the maximum entry of $(\boldsymbol{\gamma}^*, \boldsymbol{\lambda}^*)$
    $\tilde{k} \leftarrow \underset{k \in \tilde{\mathcal{K}}}{\text{argmax}}\ c_m^k(s) x_{m,\tilde{n}}^k \ \big( \sum_{\substack{j=1 \\ \exists q: f_{j,q} = \tilde{f}}}^{J} b_i^k y_{i,j}^k \big)$
    $\tilde{\mathcal{K}} \leftarrow \tilde{\mathcal{K}} \setminus \{\tilde{k}\}$
end
Step 3: Reconstruct the vector $\tilde{\boldsymbol{\alpha}}$ of size $K$, with entries indexed by $\tilde{\mathcal{K}}$ being set to 1 and 0 otherwise.
Step 4: Go to Step 1.
```

it appropriate for evaluating offline AC/FGE algorithms. In addition, the optimum solution is found by solving hard AC/FGE ILP and soft AC/FGE MILP problems with intlinprog function in MathWorks® Optimization Toolbox™. For each simulation study, 100 iterations are run.

A. Evaluation of Re-Composition Stage

First, we investigate the impact of the introduced system model, encompassing multiple input/output scenarios and enriched by dummy nodes, on fortifying chain re-composition stage. We consider NFVI 1 in Fig. 5a and 20 pairs of VNFRs, where the two VNFRs of each pair are fully correlated. VNFRs' graphs are assumed to be similar to the one in Fig. 2a, excluding v_3^1 and with $C_1, B \sim \text{Rayleigh}(1)$. From each pair, one VNFR's destination is $\bar{v}_4$, while the other one is destined to $\bar{v}_5$, with the same sources $\bar{v}_1$. We study such a scenario in three cases: 1) the VNFRs are directly considered as the input of hard AC/FGE, 2) each pair's VNFRs are combined through a chain re-composition stage so as to form a multi-output graph, similar to Fig. 2b, and 3) a dummy node is added to the multi-output graph of previous case, according to Fig. 2c. The resulting acceptance rate and revenue related to each of these cases are shown in table II. The improvements made by chain re-composition stage by employing multiple input/output model and dummy nodes can be seen in the columns corresponding to cases 2 and 3, respectively.



(a) NFVI 1.

(b) NFVI 2.

Fig. 5: The NFVI topologies considered in the simulations.

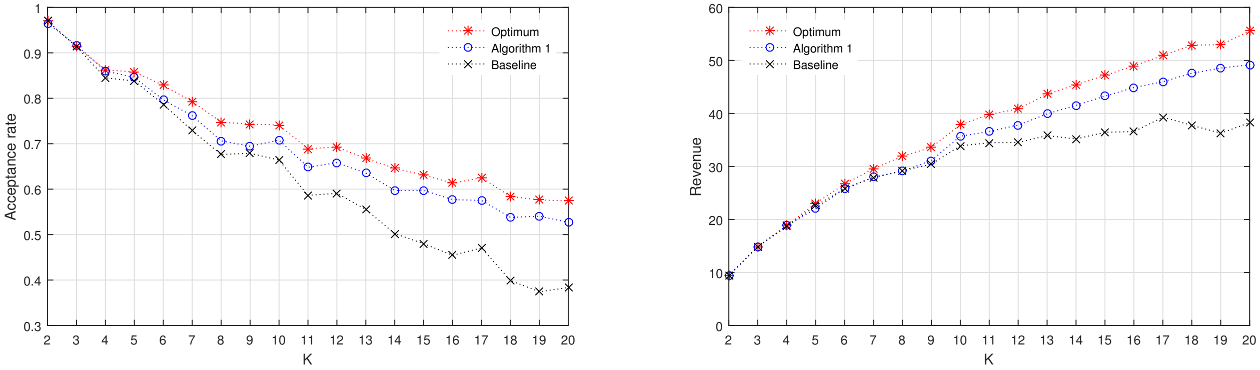
TABLE II: Effect of multiple input/output model and dummy nodes on acceptance rate and revenue through chain re-composition.

	Case 1	Case 2	Case 3
Acceptance rate	0.266	0.401	0.504
Revenue	29.998	44.996	59.995

B. Evaluation of AC/FGE Solutions

Next, in order to assess the two proposed AC/FGE methods, we consider NFVI 2 shown in Fig. 5b, which is a realization of Erdos-Renyi random graph model [39] $\mathcal{G}(n, p)$ with $n = 12$ and $p = 0.25$ as the NFVI. The processing capacities of substrate nodes and bandwidth of substrate links are assumed to be 5 units. Additionally, VNFRs are generated with respect to the Erdos-Renyi random graph model $\mathcal{G}(n, p)$ with $n \sim \text{Poisson}(1) + 2$ and $p = 0.7$. The capacities of virtual nodes and links of the VNFRs follow the distribution of $\text{Rayleigh}(1)$.

Fig. 6 pertains to the hard AC/FGE problem, and represents the performance of Algorithm 1 in terms of acceptance rate (Fig. 6a) and revenue (Fig. 6b). In Fig. 6a, the acceptance rate of AC stage is depicted, which is closely related to the obtained revenue. As expected, a downward trend in acceptance rate by increasing number of requests is evident, which is due to the limited capacity of NFVI. Moreover, as shown in Fig. 6b, total revenue is directly related to the number of input VNFRs. It can also be seen that the growth rate of revenue by number of requests is decreasing, since the NFVI approaches its saturation point. Therefore, according to Fig. 6, Algorithm 1 outperforms the baseline algorithm to solve hard AC/FGE, and the larger the number of requests is, the greater the performance improvement will be.



(a) Acceptance rate vs. number of requests

(b) Revenue vs. number of requests

Fig. 6: The revenue and acceptance rate obtained by Algorithm 1 to solve hard AC/FGE for different numbers of requests.

Fig. 7 demonstrates the effectiveness of Algorithm 2 in solving soft AC/FGE problem. Fig. 7a shows that, when the number of requests rises, the baseline algorithm's acceptance rate declines rapidly, while the acceptance rate resulting from Algorithm 2 decreases with a smaller slope. Furthermore, Fig. 7b demonstrates how the revenue obtained through Algorithm 2 is close to the one obtained by the optimum solution. However, by increasing the number of requests, the performance of the baseline algorithm falls sharply, which, in part, is due to the fact that VNF splitting/multi-path routing was not permitted. This indicates that splittable VNF and multi-path routing in VNF-FGE are of high benefit in the AC process, leading to a much larger number of admitted VNFRs.

VIII. Conclusions

In the present study, an NFV architecture was considered, where a network provider aims at maximizing its own revenue made by serving VNFRs. We presented a comprehensive system model for NFV components, i.e. services, NFVI, and NFV-MANO. Within such a framework, we investigated the problem of resource allocation, particularly the joint task of AC and VNF-FGE. Also, two variants of such a problem with respect to VNF splitting and multi-path routing were considered.

We mathematically formulated the joint task as an ILP in case of unsplittable VNF and non-permissible multi-path routing, and as a MILP, when covering splittable VNF and multi-path routing scenarios. To find the solution of AC/FGE in a tractable manner, we employed relaxation and SCA methods as well as heuristic approaches. Through simulation results, we demonstrated that such methods outperform earlier methods presented to address the AC/FGE problem.

In order to conduct further research in the area of NFV resource allocation, one may attempt to concentrate on the relation between AC and the other stages of resource allocation, thereby achieving a more comprehensive model. The study of different types of VNFs to discover how a VNF functionality is associated with its level of splittability is also a direction for future work. As a result and an extension of this study, VNFs can be classified and the problem of hybrid AC/FGE, a combination of hard and soft versions, can be investigated. In this regard, proposing a detailed model to specify and distinguish between the hardware/software requirements as well as the cost functions of hard and soft AC/FGE is of significant value. Another suggestion inspired by the present work is to employ the solution of soft AC/FGE as the process of re-planning the substrate network. For instance, depending on the statistics of input traffic, resources with greater-than-one scale factors generally lead to bottlenecks, the extent of which are directly related to the resulting scale factors. Thus, the larger the optimal scale factor of a resource is, the more additional capacity should be provisioned in it.

References

[1] Y. Li and M. Chen, "Software-defined network function virtualization: A survey," *IEEE Access*, vol. 3, pp. 2542–2553, 2015.

[2] T. Wood, K. Ramakrishnan, J. Hwang, G. Liu, and W. Zhang, "Toward a software-based network: Integrating software defined networking and network function virtualization," *IEEE Network*, vol. 29, no. 3, pp. 36–41, 2015.

[3] P. Quinn and T. Nadeau, "Problem statement for service function chaining," Internet Requests for Comments, RFC 7498, 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc7498.txt

[4] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Communications Magazine*, vol. 53, no. 2, pp. 90–97, 2015.

[5] 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 Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 236–262, 2016.

[6] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, "NFV: State of the art, challenges, and implementation in next generation mobile networks (vEPC)," *IEEE Network*, vol. 28, no. 6, pp. 18–26, 2014.

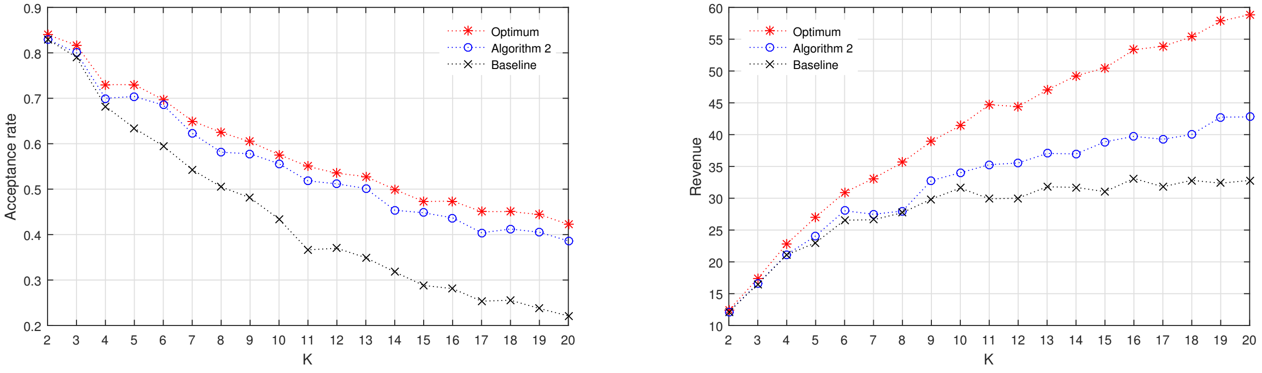
[7] ETSI Industry Specification Group (ISG) NFV, "ETSI GS NFV 002 V1.2.1: Network functions virtualisation (NFV); architectural framework," 2014. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_NFV002v010201p.pdf

[8] J. G. Herrera and J. F. Botero, "Resource allocation in NFV: A comprehensive survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 518–532, 2016.

[9] W. John, K. Pentikousis, G. Agapiou, E. Jacob, M. Kind, A. Manzalini, F. Risso, D. Staessens, R. Steinert, and C. Meirosu, "Research directions in network service chaining," in *2013 IEEE SDN for Future Networks and Services (SDN4FNS)*, 2013, pp. 1–7.

[10] S. Mehraghdam, M. Keller, and H. Karl, "Specifying and placing chains of virtual network functions," in *2014 IEEE 3rd International Conference on Cloud Networking (CloudNet)*, 2014, pp. 7–13.

[11] M. F. Bari, S. R. Chowdhury, R. Ahmed, and R. Boutaba, "On orchestrating virtual network functions," in *2015 11th International Conference on Network and Service Management (CNSM)*, 2015, pp. 50–56.



(a) Acceptance rate vs. number of requests

(b) Revenue vs. number of requests

Fig. 7: The revenue and acceptance rate obtained by Algorithm 2 to solve soft AC/FGE for different numbers of requests.

[12] A. Fischer, J. F. Botero, M. T. Beck, H. De Meer, and X. Hesselbach, "Virtual network embedding: A survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 1888–1906, 2013.

[13] N. M. K. Chowdhury, M. R. Rahman, and R. Boutaba, "Virtual network embedding with coordinated node and link mapping," in *IEEE INFOCOM 2009*, 2009, pp. 783–791.

[14] A. Belbekkouche, M. M. Hasan, and A. Karmouch, "Resource discovery and allocation in network virtualization," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 1114–1128, 2012.

[15] R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. De Turck, and S. Davy, "Design and evaluation of algorithms for mapping and scheduling of virtual network functions," in *2015 1st IEEE Conference on Network Softwarization (NetSoft)*, 2015, pp. 1–9.

[16] J. F. Riera, E. Escalona, J. Batalle, E. Grasa, and J. A. Garcia-Espin, "Virtual network function scheduling: Concept and challenges," in *2014 International Conference on Smart Communications in Network Technologies (SaCoNeT)*. IEEE, 2014, pp. 1–5.

[17] E. Amaldi, S. Coniglio, A. M. Koster, and M. Tieves, "On the computational complexity of the virtual network embedding problem," *Electronic Notes in Discrete Mathematics*, vol. 52, pp. 213–220, 2016.

[18] T. W. Kuo, B. H. Liou, K. C. J. Lin, and M. J. Tsai, "Deploying chains of virtual network functions: On the relation between link and server usage," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, 2016, pp. 1–9.

[19] T. Lukovszki and S. Schmid, "Online admission control and embedding of service chains," in *International Colloquium on Structural Information and Communication Complexity*. Springer, 2015, pp. 104–118.

[20] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding: Substrate support for path splitting and migration," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 17–29, 2008.

[21] M. C. Luizelli, L. R. Bays, L. S. Buriol, M. P. Barcellos, and L. P. Gaspary, "Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2015, pp. 98–106.

[22] H. Moens and F. De Turck, "VNF-P: A model for efficient placement of virtualized network functions," in *2014 10th International Conference on Network and Service Management (CNSM)*, 2014, pp. 418–423.

[23] B. Addis, D. Belabed, M. Bouet, and S. Secci, "Virtual network functions placement and routing optimization," in *2015 IEEE 4th International Conference on Cloud Networking (CloudNet)*, 2015, pp. 171–177.

[24] R. Riggio, A. Bradai, T. Rasheed, J. Schulz-Zander, S. Kuklinski, and T. Ahmed, "Virtual network functions orchestration in wireless networks," in *2015 11th International Conference on Network and Service Management (CNSM)*, 2015, pp. 108–116.

[25] M. Ghaznavi, A. Khan, N. Shahriar, K. Alsubhi, R. Ahmed, and R. Boutaba, "Elastic virtual network function placement," in *2015 IEEE 4th International Conference on Cloud Networking (CloudNet)*, 2015, pp. 255–260.

[26] R. Bruschi, A. Carrega, and F. Davoli, "A game for energy-aware allocation of virtualized network functions," *Journal of Electrical and Computer Engineering*, vol. 2016, p. 2, 2016.

[27] T. Taleb, A. Ksentini, and B. Sericola, "On service resilience in cloud-native 5G mobile systems," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 483–496, 2016.

[28] M. T. Beck, J. F. Botero, and K. Samelin, "Resilient allocation of service function chains," in *IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 2016, pp. 128–133.

[29] F. Carpio, S. Dhahri, and A. Jukan, "VNF placement with replication for load balancing in NFV networks," *arXiv preprint arXiv:1610.08266*, 2016.

[30] S. D'Oro, L. Galluccio, S. Palazzo, and G. Schembra, "Exploiting congestion games to achieve distributed service chaining in NFV networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 2, pp. 407–420, 2017.

[31] M. T. Beck and J. F. Botero, "Coordinated allocation of service function chains," in *2015 IEEE Global Communications Conference (GLOBECOM)*, 2015, pp. 1–6.

[32] L. Wang, Z. Lu, X. Wen, R. Knopp, and R. Gupta, "Joint optimization of service function chaining and resource allocation in network function virtualization," *IEEE Access*, vol. 4, pp. 8084–8094, 2016.

[33] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[34] J. Y. Yen, "Finding the k shortest loopless paths in a network," *Management Science*, vol. 17, no. 11, pp. 712–716, 1971.

[35] ETSI Industry Specification Group (ISG) NFV, "ETSI GS NFV-INF 004 V1.1.1: Network functions virtualisation (NFV); infrastructure; hypervisor domain," 2014. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/NFV-INF/001_099/004/01.01.01_60/gs_NFV-INF004v010101p.pdf

[36] H. M. Tseng, H. L. Lee, J. W. Hu, T. L. Liu, J. G. Chang, and W. C. Huang, "Network virtualization with cloud virtual switch," in *2011 IEEE 17th International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2011, pp. 998–1003.

[37] H. L. Bodlaender, P. Gritzmann, V. Klee, and J. v. Leeuwen, "Computational complexity of norm-maximization," *Combinatorica*, vol. 10, no. 2, pp. 203–225, 1990.

[38] T. T. De Rubira, "Norm maximization algorithm," 2011. [Online]. Available: https://web.stanford.edu/class/cme334/docs/2011-11-17-Rubira_normmax.pdf

[39] P. Erdos and A. Renyi, "On random graphs I," *Publicationes Mathematicae Debrecen*, vol. 6, pp. 290–297, 1959.

APPENDIX A
PROOF OF THEOREM 1

Proof. Let Ω_1 and Ω_2 be the feasible regions of $\mathcal{P}_1$ and $\mathcal{P}_2$, which are specified by the sets of constraints (10)–(11) and (12)–(14), respectively. As stated in the theorem, the optimal value of $\mathcal{P}_1$ equals R^*, so it is obvious that constraint (14) is implicitly considered in Ω_1, which results in $\Omega_1 \subseteq \Omega_2$.

The following remarks will be also used in the proof.

Remark 1. For any $\hat{\boldsymbol{z}} \in [0,1]^{\hat{D}}$ and $\boldsymbol{r} \in (0,\infty)^{\hat{D}}$ we have:

$$g_2(\hat{\boldsymbol{z}}) = \sum_{d=1}^{\hat{D}} r_d \hat{z}_d^2 \leqslant \sum_{d=1}^{\hat{D}} r_d \hat{z}_d = g_1(\hat{\boldsymbol{z}}),$$

and equality holds if and only if $\hat{\boldsymbol{z}} \in \{0,1\}^{\hat{D}}$.

Remark 2. For any $\check{\boldsymbol{z}} \in \{0,1\}^{\check{D}}$, $\check{\boldsymbol{z}}' \in [0,1]^{\check{D}}$, and $\delta > 0$ we have:

$$h_2(\check{\boldsymbol{z}}') = \delta \sum_{d=1}^{\check{D}} (\check{z}'_d - \frac{1}{2})^2 \leqslant \delta \sum_{d=1}^{\check{D}} (\check{z}_d - \frac{1}{2})^2 = h_2(\check{\boldsymbol{z}}),$$

and equality holds if and only if $\check{\boldsymbol{z}}' \in \{0,1\}^{\check{D}}$.

Then, we provide the proof in the following two parts.

1) We show that if $\boldsymbol{z}_1^* = (\hat{\boldsymbol{z}}_1^*, \check{\boldsymbol{z}}_1^*)$ is the optimal solution of $\mathcal{P}_1$, it is also an optimal solution of $\mathcal{P}_2$.
First, as $\Omega_1 \subseteq \Omega_2$, $\boldsymbol{z}_1^*$ is a feasible point of $\mathcal{P}_2$ as well. In addition, if $\boldsymbol{z}_2^* = (\hat{\boldsymbol{z}}_2^*, \check{\boldsymbol{z}}_2^*)$ is the optimal solution of $\mathcal{P}_2$, we have:

$$\begin{aligned} f_2(\boldsymbol{z}_2^*) &= g_2(\hat{\boldsymbol{z}}_2^*) + h_2(\check{\boldsymbol{z}}_2^*) \\ &\overset{(i)}{\leqslant} g_1(\hat{\boldsymbol{z}}_2^*) + h_2(\check{\boldsymbol{z}}_2^*) \\ &\overset{(ii)}{\leqslant} g_1(\hat{\boldsymbol{z}}_2^*) + h_2(\check{\boldsymbol{z}}_1^*) \\ &\overset{(iii)}{\leqslant} g_1(\hat{\boldsymbol{z}}_1^*) + h_2(\check{\boldsymbol{z}}_1^*) \\ &\overset{(iv)}{=} g_2(\hat{\boldsymbol{z}}_1^*) + h_2(\check{\boldsymbol{z}}_1^*) \\ &= f_2(\boldsymbol{z}_1^*), \end{aligned} \tag{22}$$

where (i) and (iv) come from Remark 1 and the fact that $\boldsymbol{z}_1^*$ is a binary vector. Remark 2 and $\boldsymbol{z}_1^*$ being binary lead to (ii), and constraint (14) also results in (iii). Therefore, $\boldsymbol{z}_1^* \in \Omega_2$ and $f_2(\boldsymbol{z}_2^*) \leqslant f_2(\boldsymbol{z}_1^*)$ prove that $\boldsymbol{z}_1^*$ is an optimal solution of $\mathcal{P}_2$.

2) Next, we show that if $\boldsymbol{z}_2^* = (\hat{\boldsymbol{z}}_2^*, \check{\boldsymbol{z}}_2^*)$ is the optimal solution of $\mathcal{P}_2$, it is also an optimal solution of $\mathcal{P}_1$.
According to the assumption that Ω_1 is not empty, there is at least an optimal solution $\boldsymbol{z}_1^* = (\hat{\boldsymbol{z}}_1^*, \check{\boldsymbol{z}}_1^*)$, which is also a feasible point of $\mathcal{P}_2$ (since $\Omega_1 \subseteq \Omega_2$). In addition the optimality of $\boldsymbol{z}_2^*$ for $\mathcal{P}_2$ leads to:

$$f_2(\boldsymbol{z}_1^*) \leqslant f_2(\boldsymbol{z}_2^*),$$

which, coupled with the fact that (22) is still held true, turns (i), (ii), and (iii) into equalities, and shows that:

$$g_2(\hat{\boldsymbol{z}}_2^*) = g_1(\hat{\boldsymbol{z}}_2^*), \tag{23}$$

$$h_2(\check{\boldsymbol{z}}_2^*) = h_2(\check{\boldsymbol{z}}_1^*), \tag{24}$$

$$g_1(\hat{\boldsymbol{z}}_2^*) = g_1(\hat{\boldsymbol{z}}_1^*). \tag{25}$$

Taking (23) and Remark 1 as well as (24) and Remark 2 into account, it is immediately concluded that $\hat{\boldsymbol{z}}_2^*$ and $\check{\boldsymbol{z}}_2^*$ are binary vectors. As a result, $\boldsymbol{z}_2^*$ is a binary vector, and belongs to Ω_1, considering the fact that $\boldsymbol{z}_2^* \in \Omega$. (25) is also equivalent to $f_1(\boldsymbol{z}_2^*) = f_1(\boldsymbol{z}_1^*)$, combined with $\boldsymbol{z}_2^* \in \Omega_1$, confirms that $\boldsymbol{z}_2^*$ is an optimal solution for $\mathcal{P}_1$, which concludes the proof.

□

Algorithm A: NORM MAXIMIZATION SCA [38]

```
function NormMax($\tilde{\boldsymbol{U}}, \tilde{\boldsymbol{v}}, \epsilon$)
Step 1: Find max-volume ellipsoid $\mathcal{E} = \{\boldsymbol{Q}\boldsymbol{u} \leqslant \boldsymbol{r} | \|\boldsymbol{u}\|_2 \leqslant 1\}$ inscribed inside $\{\tilde{\tilde{\boldsymbol{z}}} | \tilde{\boldsymbol{U}}\tilde{\tilde{\boldsymbol{z}}} \leqslant \tilde{\boldsymbol{v}}\}$, by solving the following optimization, with $\boldsymbol{Q}^*$ and $\boldsymbol{r}^*$ being the optimum solutions:
    $\max_{\boldsymbol{Q} \in \mathbb{R}^{\tilde{D} \times \tilde{D}}, \boldsymbol{r} \in \mathbb{R}^{\tilde{D}}} \det(\boldsymbol{Q})$
    s.t.  $\|\boldsymbol{Q}\tilde{\boldsymbol{u}}_e\|_2 \leqslant \tilde{v}_e, \quad \forall e \in \{1, 2, \ldots, \tilde{E}\}.$
Step 2: Solve the original norm maximization problem over feasible region $\mathcal{E}$ (as it is a non-convex optimization, the dual problem, which is convex, will be solved), and the optimum solutions are $\gamma^*$ and $\lambda^*$:
    $\max_{\gamma, \lambda} \gamma$
    s.t.  $\lambda \geqslant 0,$
          $\begin{bmatrix} -\boldsymbol{I} + \lambda \boldsymbol{Q}^{*-1}\boldsymbol{Q}^{*-1} & -\lambda \boldsymbol{Q}^{*-1}\boldsymbol{Q}^{*-1}\boldsymbol{r}^* \\ -\lambda \boldsymbol{r}^{*T}\boldsymbol{Q}^{*-1}\boldsymbol{Q}^{*-1} & \lambda \boldsymbol{r}^{*T}\boldsymbol{Q}^{*-1}\boldsymbol{Q}^{*-1}\boldsymbol{r}^* - \lambda - \gamma \end{bmatrix} \succeq 0.$
$\tilde{\tilde{\boldsymbol{z}}}^* \leftarrow (-\boldsymbol{I} + \lambda^*\boldsymbol{Q}^{*-1}\boldsymbol{Q}^{*-1})^\dagger \lambda^* \boldsymbol{Q}^{*-1}\boldsymbol{Q}^{*-1}\boldsymbol{r}^*$
Step 3: Construct supporting half-space $\{\tilde{\tilde{\boldsymbol{z}}} | \boldsymbol{g}^T\tilde{\tilde{\boldsymbol{z}}} \leqslant \boldsymbol{h}\}$ of $\mathcal{E}$ at $\tilde{\tilde{\boldsymbol{z}}}^*$:
$f(\tilde{\tilde{\boldsymbol{z}}}) \triangleq$
$\tilde{\tilde{\boldsymbol{z}}}^T\boldsymbol{Q}^{*-1}\boldsymbol{Q}^{*-1}\tilde{\tilde{\boldsymbol{z}}} - 2\boldsymbol{r}^{*T}\boldsymbol{Q}^{*-1}\boldsymbol{Q}^{*-1}\tilde{\tilde{\boldsymbol{z}}} + \boldsymbol{r}^{*T}\boldsymbol{Q}^{*-1}\boldsymbol{Q}^{*-1}\boldsymbol{r}^* - 1$
$\tilde{\boldsymbol{U}} \leftarrow \begin{bmatrix} \tilde{\boldsymbol{U}} \\ -\nabla f(\tilde{\tilde{\boldsymbol{z}}}^*)^T \end{bmatrix}$
$\tilde{\boldsymbol{v}} \leftarrow \begin{bmatrix} \tilde{\boldsymbol{v}} \\ -\nabla f(\tilde{\tilde{\boldsymbol{z}}}^*)^T \tilde{\tilde{\boldsymbol{z}}}^* \end{bmatrix}$
Step 4: Stop if the major axis of $\mathcal{E}$ is less than a certain amount:
if $2 \times$ (largest eigenvalue of $\boldsymbol{Q}^*$) $< \epsilon$ then
    return $\tilde{\tilde{\boldsymbol{z}}}^*$
else
    Go to Step 1.
end
end function
```

APPENDIX B
NORM MAXIMIZATION SCA METHOD

The method proposed in [38] has been described in Algorithm A to solve the following problem:

$$\begin{aligned} \max_{\tilde{\tilde{\boldsymbol{z}}} \in \mathbb{R}^{\tilde{D}}} \quad & \|\tilde{\tilde{\boldsymbol{z}}}\|_2 \\ \text{s.t.} \quad & \tilde{\boldsymbol{U}}\tilde{\tilde{\boldsymbol{z}}} \leqslant \tilde{\boldsymbol{v}}, \end{aligned} \tag{26}$$

where $\tilde{\boldsymbol{U}} \in \mathbb{R}^{\tilde{E} \times \tilde{D}}$ and $\tilde{\boldsymbol{v}} \in \mathbb{R}^{\tilde{D}}$ are such that the feasible region is bounded with non-empty interior. We also assume $\tilde{\boldsymbol{v}} = (\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_{\tilde{E}})$ and:

$$\tilde{\boldsymbol{U}} = \begin{bmatrix} \tilde{\boldsymbol{u}}_1^T \\ \vdots \\ \tilde{\boldsymbol{u}}_{\tilde{E}}^T \end{bmatrix},$$

where $\tilde{\boldsymbol{u}}_e \in \mathbb{R}^{\tilde{D}}$, $\forall e \in \{1, 2, \ldots, \tilde{E}\}$.

APPENDIX C
PROOF OF PROPOSITION 1

Proof. Through quantization process, the binary constraint $\bar{\boldsymbol{z}} \in \{0,1\}^{\tilde{D}}$ of (9) is clearly true. It remains to demonstrate that $\bar{\boldsymbol{z}}$ belongs to the feasible region meeting constraints $(\mathrm{C2}, \mathrm{C3}, \mathrm{C5}, \mathrm{C6})$.

First, the rounding-down procedure is shown to preserve the feasibility of $\bar{z}$ in (9). As $\tilde{z}$ is contained in the feasible region $(\tilde{\mathrm{C}}2, \tilde{\mathrm{C}}3, \mathrm{C}6)$, it is obviously a feasible point of $(\mathrm{C}2, \mathrm{C}3, \mathrm{C}6)$, even by having some of the elements rounded down. As for C5, if at least one of the elements $x_{m,n}^k, m \in \{m : v_m^k \in e_i^k\}$ is reset to zero, it means $\hat{y}_{i,n}^k$ is also less than $1-\epsilon$, and has to be reset to zero. Thus, C5 still holds and, after rounding down the elements less than $1-\epsilon$ to 0, $\bar{z}$ is in the feasible set $(\mathrm{C}2, \mathrm{C}3, \mathrm{C}5, \mathrm{C}6)$.

Next, we prove that the rounding-up procedure also respects the constraints of (9). If all the elements of $\tilde{z}$ need to be rounded up to 1, in the worst case, left-hand sides of inequalities $(\tilde{\mathrm{C}}2, \tilde{\mathrm{C}}3)$ are multiplied by $\frac{1}{1-\epsilon}$. Even in such a case, since the right-hand sides of $(\mathrm{C}2, \mathrm{C}3)$ are $\frac{1}{1-\epsilon}$ times as much as the one of $(\tilde{\mathrm{C}}2, \tilde{\mathrm{C}}3)$, the inequalities $(\mathrm{C}2, \mathrm{C}3)$ are met for $\bar{z}$. In addition, if $\hat{y}_{i,n}^k$ in C5 is rounded up to 1, all the elements $x_{m,n}^k, m \in \{m : v_m^k \in e_i^k\}$ are inevitably greater than or equal to $1-\epsilon$, and should be rounded up as well, which means C5 is still satisfied after the rounding-up process. Finally, C6 and $0 \leqslant \epsilon < 0.5$ lead to the fact that, among elements $x_{m,n}^k, n \in \{1, 2, \ldots, N\}$, at most one element is greater than or equal to $1-\epsilon$. Hence, after the quantization process is done, not more than one element is set to 1 and the others are reset to 0, resulting in C6 being still held true.

Therefore, the quantization procedure ensures the feasibility with respect to the constraints of (9), i.e. $(\bar{z} \in \{0,1\}^{\tilde{D}}, \mathrm{C}2, \mathrm{C}3, \mathrm{C}5, \mathrm{C}6)$. □

Mohammad Ali Tahmasbi Nejad (S'14) received the B.Sc. degree in electrical engineering and the M.Sc. degree in communications systems both from Sharif University of Technology, Tehran, Iran, in 2014 and 2016, respectively. Since graduation, he has been a research assistant with the Department of Communication Technologies, Iran Telecommunication Research Center, and has been working on the key technologies of next generation 5G wireless communication systems. His research interests include wireless communication, mobile network optimization, content distribution, network coding, network function virtualization, and software-defined networking.

Saeedeh Parsaeefard (S'09–M'14) received the B.Sc. and M.Sc. degrees from Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran, in 2003 and 2006, respectively, and the Ph.D. degree in electrical and computer engineering from Tarbiat Modares University, Tehran, Iran, in 2012. She was a Post-Doctoral Research Fellow with the Telecommunication and Signal Processing Laboratory in the Department of Electrical and Computer Engineering at the McGill University, Canada. From November 2010 to October 2011, she was a Visiting Ph.D. Student with the Department of Electrical Engineering, University of California, Los Angeles, CA, USA. She is currently a research assistant in the Department of Electrical and Computer Engineering at the McGill University, Canada and a faculty member in Iran Telecommunication research center. Her current research interests include the resource management in software-defined networking, internet of things and the fifth generation of wireless networks as well as applications of robust optimization theory and game theory on the resource allocation and management in wireless networks.

Mohammad Ali Maddah-Ali (S'03–M'08) received the B.Sc. degree from Isfahan University of Technology, and the M.A.Sc. degree from the University of Tehran, both in electrical engineering. From 2002 to 2007, he was with the Coding and Signal Transmission Laboratory (CST Lab), Department of Electrical and Computer Engineering, University of Waterloo, Canada, working toward the Ph.D. degree. From 2007 to 2008, he worked at the Wireless Technology Laboratories, Nortel Networks, Ottawa, ON, Canada. From 2008 to 2010, he was a post-doctoral fellow in the Department of Electrical Engineering and Computer Sciences at the University of California at Berkeley. Then, he joined Bell Labs, Holmdel, NJ, as a communication research scientist. Recently, he started working at Sharif University of Technology, as a faculty member. Dr. Maddah-Ali is a recipient of NSERC Postdoctoral Fellowship in 2007, a best paper award from IEEE International Conference on Communications (ICC) in 2014, the IEEE Communications Society and IEEE Information Theory Society Joint Paper Award in 2015, and the IEEE Information Theory Society Joint Paper Award in 2016.

Toktam Mahmoodi received the B.Sc. degree in electrical engineering from the Sharif University of Technology, Iran, and the Ph.D. degree in telecommunications from Kings College London, U.K. She was a Visiting Research Scientist with F5 Networks, San Jose, CA, in 2013, a Post-Doctoral Research Associate with the ISN Research Group, Electrical and Electronic Engineering Department, Imperial College from 2010 to 2011, and a Mobile VCE Researcher from 2006 to 2009. She has also worked in mobile and personal communications industry from 2002 to 2006, and in an R&D team on developing DECT standard for WLL applications. She has contributed to, and led number of FP7, H2020 and EPSRC funded projects, advancing mobile and wireless communication networks. Toktam is currently with the academic faculty of Centre for Telecommunications Research at the Department of Informatics, Kings College London. Her research interests include 5G communications, network virtualization, and low latency networking.

Babak Hossein Khalaj received the B.Sc. degree in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 1989, and the M.Sc. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 1993 and 1996, respectively. He has been with the pioneering team at Stanford University, where he was involved in adoption of multi-antenna arrays in mobile networks. He joined KLA-Tencor in 1995, as a Senior Algorithm Designer, working on advanced processing techniques for signal estimation. From 1996 to 1999, he was with Advanced Fiber Communications and Ikanos Communications. Since then, he has been a Senior Consultant in the area of data communications, and a Visiting Professor at CEIT, San Sebastian, Spain, from 2006 to 2007. He has co-authored many papers in signal processing and digital communications. He holds two U.S. patents and the recipient of Alexander von Humboldt Fellowship from 2007 to 2008. He was the Co-Editor of the Special Compatibility Standard Draft for ANSIT1E1 Group from 1998 to 1999.