Flexible Compositions for the Virtual Network Function Chain Placement in Online Environments

Samuel M. A. Araújo
Department of Computer Science
Federal University of Minas Gerais
Belo Horizonte, Brazil
Email: smaa@dcc.ufmg.br

Fernanda S. H. de Souza
Department of Computer Science
Federal University of São João del-Rei
São João del-Rei, Brazil
Email: fsumika@ufsj.edu.br

Geraldo R. Mateus
Department of Computer Science
Federal University of Minas Gerais
Belo Horizonte, Brazil
Email: mateus@dcc.ufmg.br

Abstract—**Network Functions Virtualization is gaining significant attention from researchers for reducing the costs of Telecommunication Service Providers (TSP) and giving more flexibility to market demands. Traditionally, works dealing with virtual network function placement and chaining problem receive the requests with a fixed chaining composition. In an online environment with substrate network resources partially consumed, such fixed chaining composition, although being optimal for the client, may be less profitable or impossible to be mapped by the service provider. In this case, an alternative chain composition, even with eventually higher bandwidth expenditure, may potentially have a better adjustment on the residual substrate network. This work proposes to give more autonomy to the TSP through a set of alternative chaining compositions. Such action includes flexibility in the choice of the structure that best fits on the residual network. In this model, the client is responsible to provide chaining alternatives (attending the same network service) to the TSP. The simulations show that the composition selection approach generates increases in profit and acceptance rate when compared to traditional models.**

Index Terms—**Network function virtualization, virtual network functions, resource allocation, service function chaining, chain composition.**

I. INTRODUCTION

Firewall, Web Proxy, Video Transcode/Encode, Load balance, Caching, audio, and video streaming, are examples of Network Functions (NF) appliances have become much-needed on the internet today. As the internet was not originally designed to deliver this different and massive volume of functions, a challenging task begins to form, the attendance most diverse functions, with quality, security, and low expenses with services implementation and operation [1]–[4].

Among the recent proposals for innovation and improvement of the NFs, the application of virtualization mechanisms has been gaining prominence. With Network Function Virtualization (NFV), it becomes possible to implement new and different NFs, emulated via software on general-purpose hardware. This technique allows NF not only to run on proprietary machines[1] (middleboxes) but also on generic machines suitable to virtualize the NFs requested. In this context, the Virtual Network Functions (VNFs) emerge [5]–[8].

Considering the concept of NFV networks, NFs can potentially be virtualized on any unspecified generic devices distributed by the Substrate Network (SN). These locations are called Points Of Presence (N-PoPs). By definition, a Network Service (NS) offered by an operator consists in a full end-to-end functionality that is delivered using one or more VNFs (e.g., firewall, IDS, proxy, for security purposes) [1], [4], [9]. According to The European Telecommunications Standards Institute (ETSI), a set of order VNFs, or partially ordered, demanded by an NS is called the Service Function Chain (SFC) or VNF Forwarding Graph (VNF-FG) [10]. It is important to note that several and different SFCs can satisfy the same NS since the order of the VNFs is not always fixed, i.e., some VNFs are dependent in their sequence while others are shown to be flexible [11].

In the network business model, the Infrastructure Providers (InPs) are responsible for implementing and managing the SN physical resources. Another entity of this interaction model is the TSPs, which lease the devices of the InPs to running VNFs. It is the TSPs who have to manage the NFs chaining to build services for end users [8]. One of the challenges encountered by TSPs is to efficiently map the SFC requests on SN, where a limited capacity of available resources should be considered. [5], [12].

One of the problems addressed as a reference in this article is known as network positioning and chaining (VNF-PC) [5], [13], [14]. Into VNF-PC it is required to allocate the instances of VNFs and place the clients (end-points) in a viable region of the SN, besides providing an ordered routing between the pairs of instantiated components. The difficulties for solving the VNF-PC are related to combinatorial nature, being the problem optimization version belongs to the NP-hard class [15]–[17].

Figure 1 shows two SFC requests mapped on SN of non-proprietary machines, prepared to virtualize any network function. In the VNF-PC, the VNFs requested can potentially be instantiated in any N-PoP present in the SN, provided that resources are available. However, according to the literature, one of the ways to generate a reduction in the deployment and operation cost is to try grouping the VNFs in the same N-PoP, reducing instantiation points and mitigating expenses [5], [15], [18].

---

[1] Proprietary machines are typically expensive, low reuse and high coupling between hardware and software.

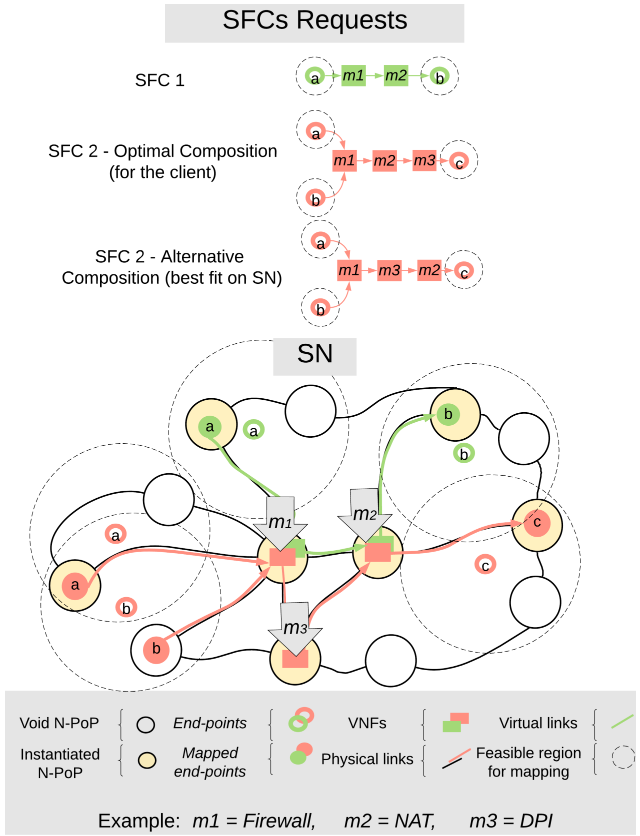

Fig. 1. Two SFCs mapped on SN, in this case, the SFC 2 was mapped with an alternative composition.

The initial challenge addressed in this work is to solve the VNF-PC, grouping the VNFs when it is profitable, encouraging sharing of instances, and generate a route that passes in an ordered way by the VNFs connecting the end-points. To Address this problem, we developed a model in Integer Linear Programming (ILP), which performs not only the positioning and chaining of the VNFs but also a selection of the SFC composition that best fit on the residual SN. In this case, we assume that the client necessity to inform the provider a set of SFCs alternatives that satisfy the NS. We assume that this action promotes cost savings in the implementation and operation of NFV networks, as well as minimizing losses with poorly planned connections.

Another challenge explored in this work is associated with the dynamics of the requisitions arrival to be processed, which usually varies between online or offline models [8], [13]. In an offline scenario, the mapping of the entire set of SFCs is performed at the same time. In this case, the mapping algorithm thoroughly knows the topologies and requirements of each SFC and can benefit from this fact. In another way, given the dynamic requirements in the NFV context, there is a need for resource allocation proposals that can find solutions online [8]. The characteristics of the online scenario diverge from the offline scenario, mainly due to the arrival of SFC requests in an unknown form. In this situation, the algorithm does not know any features about the topology, lifetime and required requirements, being this challenge utilized in this work.

In the online mapping context, the TSP must map each SFC request on the SN at the time of its arrival, where each SFC request has a duration time. Thus, it is up to the TSP to implement a set of management mechanisms for the allocation of resources of the SFC when it is active, and the restoration of resources when it completes its lifetime. [13]. Figure 2 shows an example of the arrival of SFCs in an online environment. At this rate, the flow of SFC requests is processed according to the arriving order $Arrival = \{\{SFC1\}, \{SFC2\}, \{SFC3\}, \{SFC4\}, \{SFC5\}\}$. In this case, the mapping algorithm performs five approach executions, one for each SFC.

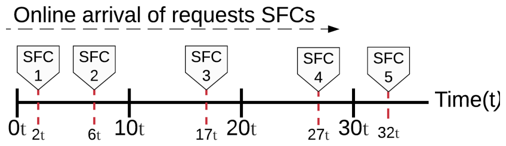

Fig. 2. Example of five SFCs arrivals in an online environment.

To perform resource allocation for a given NS, the TSP receives a composition of VNFs to be instantiated and chained on SN. At this instant, there may be a dispute of interests among the client and the TSP. The client tends to provide a predefined SFC with a rigid chain that is more beneficial to them but which neglects the residual state of the SN. However, some times, the TSP may have troubles in mapping a specific rigid composition of VNFs. In practice, some NFs have explicit dependencies (precedences) have to be considered (e.g., encryption need occur before the decryption), while others are flexible and admit different chaining possibilities. (e.g., there is no explicit dependency between a proxy server and a WAN optimizer) [10], [19]–[21]. The main contribution of this article is to show that even an optimal SFC [2] from the clients' point of view, it may not be optimal when mapped on the residual SN, and can be deprecated concerning other alternatives SFCs.

Given the potential non-dependence between some VNFs, it is plausible to propose a flexibilization of SFC composition, i.e., we propose to give TSP more compositions options to serve an NS. In this transaction, considering the dependencies between some VNFs and exploring possible non-dependencies, an NS can be served by a $\mathcal{T}$ set of SFCs. This proposed variation is called in this article as Network Function Placement and Chaining with Composition Selection (VNF-SPC). The VNF-SPC adds to the VNF-PC the step of selecting in a set $\mathcal{T}$ of SFCs provided by the client, the composition that is most beneficial for the mapping on the residual SN.

Figure 3 illustrates the set $\mathcal{T}$ of alternative compositions for mapping SFC 2, and the best composition option (alternative

---

[2] An optimal composition is an NP-hard problem and is obtained solving the SFC Composition problem, in which a function is optimized to produce the best chain that serves an NS.

2) considering the residual SN with resources already occupied by SFC 1 (Figure 1). Also in Figure 3, for the definition of composition alternatives, some rules of dependence among some VNFs are considered. In the example in Figure 3, consider that to be viable in the client application, the m_1 function needs to precede m_3. A point to be considered is that that the order of chaining of the functions can cause changes in bandwidth required between the VNFs (shown in Table I).

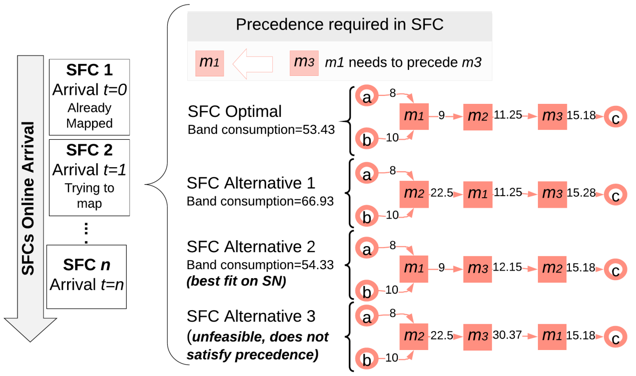

Fig. 3. Illustration of Online SFCs arrival, precedences among VNFs, and possible alternative mapping compositions for SFC 2, on the SN of Figure 1.

TABLE I
EXAMPLE OF FUNCTIONS DEMANDED BY SFC 2 FROM FIGURE 3.

Type	CPU demand	Storage demand	Bandwidth demand
m1	1 vCPU	250 GB	decreases flow 50%
m2	5 vCPU	15 GB	increases flow 25%
m3	4 vCPU	1 GB	increases flow 35%

Given the challenge of efficiently mapping SFCs, many factors need to be investigated in future works. Particularly in this article, we focus on maximizing the profitability of TSPs; for that, we proposed the reduction of expenses with inadequately planned connections and instantiations of VNFs. Another research point of this article defined by the question: Is it necessary to solve the SFC Composition Problem (SFCC) in its optimally to generate an optimal mapping of an NS? This research question is based on the fact that the resolution of the SFCC is computationally expensive.

The remainder of this paper is organized as follows. In Section II, we present a short review of the literature. In Section III, the formal description of the problem is presented together with an ILP model. Section IV presents the performance evaluation of our proposed approach. Finally, in Section V, we conclude the paper with final remarks and discusses future works.

II. RELATED WORK

The literature is crescent in terms of optimization on NFV networks. However, most of the works deal separately with the problems of Composition, Placement, and Chaining of VNFs [6]. We review these works in this section, but in our investigation, we did not find articles that address this space associated with the integration and analysis of these sub-problems.

One of the main challenges for the NFV networks deployment is the allocation of resources demanded by clients on those offered by InPs, which are usually expensive and limited. This problem is known as the NFV Resource Allocation Problem (NFV-RA) [6], [10], [20], [21]. The NFV-RA can be segmented into some sub-problems, being two of them belonging to the NP-Hard class and related to the investigation proposed in this work, they are:

***(i)* SFC Composition Problem** (SFCC) - Corresponds to the generation of the chaining composition of the VNFs that will attend the NS. In this case, the order of chaining of the functions may be subject to the precedence of VNFs and imply in lower or greater demand for resources (Figure 3). As an output, this treatment generates the SFC that will be processed later in the mapping sub-problem [10], [19]–[21].

***(ii)* SFC Embedding** (SFCE or VNF-PC) - This problem performs the instantiation, placement, and chaining of the virtual devices requested by the SFC on SN. In this case, the challenge is to optimize the allocation of SN resources to attend NS [2], [5], [11], [14], [15], [17].

The authors of [10], regard the SFCC to generate an SFC composition with minimum demands of bandwidth and subject to precedence constraints in the chain. In the approaches proposed in [10], the SFC originates in a single end-point; it passes through a set of intermediary VNFs instances; at some point, if profitable, one can split the resultant traffic into different paths; completing the sequencing of the composition in one or more end-points. The work [10] addresses the SFC composition through ILP together with an increased graph concept.

Due to the combinatorial nature of the SFCC, the authors of [20] present a heuristic for SFCC, which uses the concept of increase graphs, similar to the work of [10]. This proposal is valid by decrease the time for generation of SFC graph, which can be prohibitive in instances of medium and large size. The heuristic proposed in [20] generates an initial solution through a greedy principle of lower bandwidth consumption and submits this initial solution to a local search with Tabu List concepts.

In another treatment, the SFCC and VNF-PC subproblems can be resolved in a joint way. In this case, the chain composition and mapping can be optimized in a coordinated way, but that such action tends to raise the complexity of the model treatment and tends to be inefficient in practical terms. A proposal in this line is presented by the authors of [21], using a recursive heuristic called CoordVNF. The CoordVNF maps the VNFs recursively one by one, and when a VNF cannot be mapped, the approach performs a backtracking step returning to the last position with a valid VNF. From this point, the CoordVNF defines a new alternative for mapping the previously rejected VNF. The heuristic CoordVNF aims to minimize bandwidth usage of SN. Notwithstanding, the work of [21] does not analyze the impact of optimal or not SFCs in its mapping. In contrast to [21], our work shows that it is not necessary to solve the SFCC in optimality to produce good results in the VNF-PC.

The initial treatments applied to the NFV networks considered only the VNF Positioning (VNF-P) in servers distributed by the SN. The purpose of this variation is to position the instances of VNFs needed to serve the various clients and their demands already present in the N-PoPs [17]. As the objective function, the work proposed by [17] aims to minimize the distance between the clients and the virtual functions instances, and the costs of configuring the required VNFs. For this action, the authors of [17] make use of integer programming. Furthermore, [17] proposes a study that shows that VNF-P, even in the simplest version of the problem where there are no capacity constraints, combines two classical NP-hard problems, the Facility Location problem, and the Generalized Assignment Problem. In an evolution process of the VNF-P problem, were added the constraints of instantiating the end-points into a feasible position, and generating an ordered chain between the pairs of instantiated VNFs. To attend these new demands and constraints, the VNF-PC was formulated, such an approach is discussed in [5], [14], [22].

Work proposed by [5] presents an ILP model and a heuristic to resolve the VNF-PC. In both approaches, the objective function reduces the number of network functions instantiated. The heuristic proposed by [5] executes the ILP itself to perform the iterations. Each iteration is done with a time limit, at the end of each iteration the best solution is preserved, and the number of functions instantiated becomes a restriction to the next iteration.

The authors of [14] present a multi-objective heuristic called MO-VNFCP, for the treatment of the VNE-PC problem. MO-VNFCP uses Pareto frontier to find the best solution among the objectives: minimize the total delay, the number of hops, the number of functions instantiated and CPU usage. Similarly to authors of [5] and [14], in our work, we minimize the number of hops and functions instantiated, but we do that action an enveloped in a unique metric that maximizes profit based on resource sharing. In our proposal, the more hops are carried out, and more instantiated functions, the higher is the operational cost associated.

In [22], an approach is presented for the mapping of SFCs using ILP in the context of the mobile network. The authors of [22] emphasize that in the mobile networks scenario is characterized by the fact that network topologies are not previously known, similar to the definition of online adopted in this work. In [22] processing, storage, and switching capacity constraints are applied, in addition to restrictions that not every N-PoP can instantiate a given VNF. The objective function applied to minimizes the costs generated by the physical resources used.

III. THE NETWORK FUNCTION PLACEMENT AND CHAINING WITH COMPOSITION SELECTION

In this Section, we describe the VNF-SPC problem and introduce our proposed solution. Next, we formalize it as an ILP model.

A. Problem Overview

The VNF-SPC problem can be thought of in the realization of 3 stages with success. In the approach presented here, these stages are done together, they are:

(i) The SFC composition selection stage, as seen the client must provide for the TSP a set of possible SFCs to serve a determined NS. It is assigned to the TSP determine among the composition alternatives the one that best fits residual SN.

(ii) The stage of assigning instances, in this stage, the TSP must define the points of the SN that will receive a VNF instance in order to serve the required SFC.

(iii) The stage of SFC placement and chaining, in this stage the TSP must assign the VNFs required by SFC on the defined instances in phase ii, and interconnect the requested VNFs in an orderly way, respecting the precedence of VNFs.

B. Adopted notation and model description

The VNF-SPC can be modeled through ILP as follows: Let $G = (N, L)$ be a weighted directed graph representing the SN, where N is the set of N-PoPs (physical nodes), each N -PoP $i \in N$ has a maximum CPU capacity CPU_i and a maximum storage capacity ST_i. Each physical node $i \in N$ also has a maximum number of instantiations VM_i supported, plus a geographic positioning (x_i, y_i) determined. Let L is the set of directed physical links that connect the N-PoPs $i \in N$, so that each link $(i, j) \in L$ has a capacity of maximum band BW_{ij}. Because it is a directed graph, BW_{ij} is not necessarily equal to BW_{ji}.

Let F defined as the set of different network functions existing (e.g., NAT, DPI, Firewall, etc.). Each function $m \in F$ can be potentially virtualized on each N-PoP $i \in N$, according to the demand requirement of each VNF belonging to the SFCs. To be instantiated, each function $m \in F$ has a unit cost involved, represented by η^m, to be paid by TSP for different instances used in different N-PoPs $i \in N$.

The set $\mathcal{T}$ represents the group of different SFCs compositions for a given NS. Whether $|\mathcal{T}| = 1$, only one SFC composition attends the NS of the request, and if $|\mathcal{T}| > 1$ exist alternative compositions that must be evaluated on SN. Each $\tau \in \mathcal{T}$ composition is a chain of already defined VNFs, with the precedence of VNFs, and an end-points concatenation, which must be routed according to a sequence established by the client.

Each composition $\tau \in \mathcal{T}$ is represented as a weighted directed graph $G^\tau = (N^\tau, L^\tau)$, be $N^\tau = \{N^\tau_{end} \cup N^\tau_{vnf}\}$ the set of virtual nodes, where N^τ_{end} is the set of terminal end-points and N^τ_{vnf} the set of VNFs to be configured. Each $k \in N^\tau_{end}$ has a preferred geographical position for the mapping (x_k, y_k). Each $k \in N^\tau_{vnf}$ requires a CPU capacity cpu^τ_k and a storage capacity st^τ_k. Be L^τ the set of virtual links that make the virtual topological connection oriented between the different virtual nodes $k \in (N^\tau_{end} \cup N^\tau_{vnf})$. Each link $(k, l) \in L^\tau$ demands bandwidth capacity bw^τ_{kl} required for the mapping. Finally, each SFC request has a maximum end-to-end mapping delay to be respected, given by SFC_{dl}, an input time SFC_{te} and a time duration SFC_{td}.

A solution to the VNF-SPC consists of the feasible mapping of exactly an alternative $\tau \in \mathcal{T}$ composition of the SFC request, given by mapping $f : G^\tau \rightarrow G$, always respecting the residual conditions of SN resources and the integral demands of each incoming SFC. This mapping action is composed of the stages of selection composition, instantiation, mapping, and chaining of VNFs and end-points.

C. ILP Model

1) Variables: The ILP model for VNF-SPC operates with a set $\mathcal{T}$ of compositions. In order to model this problem, let the decision variables be:

- $y^\tau \in \{0,1\}$, indicates whether SFC composition $\tau \in \mathcal{T}$ is mapped $(y^\tau = 1)$ or not $(y^\tau = 0)$, corresponds to the composition selection (phase i).
- $w_{mi} \in \{0,1\}$, indicates whether an instance of the function m is assigned to the N-PoP i $(w_{mi} = 0)$ or not $(w_{mi} = 0)$, corresponds to the assignment of instance (phase ii).
- $z_{ik}^\tau \in \{0,1\}$, indicates whether virtual node $k \in N^\tau$ of the composition $\tau \in \mathcal{T}$ is assigned[3] on the N-PoP i $(z_{ik}^\tau = 1)$ or not $(z_{ik}^\tau = 0)$, corresponds to mapping the virtual node on a physical node (phase iii).
- $x_{ij}^{\tau kl} \in \{0,1\}$, indicates whether virtual link $(k,l) \in L^\tau$ is mapped[4] on the physical link $(i,j) \in L$ $(x_{ij}^{\tau kl} = 1)$ or not $(x_{ij}^{\tau kl} = 0)$, corresponds to the routing oriented of a virtual link on a physical link (phase iii).

2) Constraints: The proposed formulation for the VNF-SPC resolution is subject to the following sets of constraints:

$$\sum_{i \in N} z_{ik}^\tau \geq y^\tau, \forall k \in N_{vnf}^\tau, \forall \tau \in \mathcal{T} \tag{1}$$

$$\sum_{i \in N} z_{ik}^\tau a_{ik}^\tau \geq y^\tau, \forall k \in N_{end}^\tau, \forall \tau \in \mathcal{T} \tag{2}$$

$$\sum_{\tau \in \mathcal{T}} y^\tau \leq 1 \tag{3}$$

$$\sum_{k \in N_{vnf}^\tau : tipo(k) = tipo(m)} z_{ik}^\tau \leq w_{mi}, \quad \forall m \in F, \forall i \in N, \forall \tau \in \mathcal{T} \tag{4}$$

$$\sum_{k \in (N_{end}^\tau \cup N_{vnf}^\tau)} z_{ik}^\tau \leq 1, \forall i \in N, \forall \tau \in \mathcal{T} \tag{5}$$

$$\sum_{k \in N_{vnf}^\tau} z_{ik}^\tau cpu_k^\tau \leq CPU_i, \forall i \in N, \forall \tau \in \mathcal{T} \tag{6}$$

$$\sum_{k \in N_{vnf}^\tau} z_{ik}^\tau st_k^\tau \leq ST_i, \forall i \in N, \forall \tau \in \mathcal{T} \tag{7}$$

$$\sum_{m \in F} w_{mi} \leq VM_i, \quad \forall i \in N \tag{8}$$

$$\sum_{(k,l) \in L^\tau} x_{ij}^{\tau kl} bw_{kl}^\tau \leq BW_{ij}, \forall (i,j) \in L, \forall \tau \in \mathcal{T} \tag{9}$$

$$\sum_{(i,j) \in L} \sum_{(k,l) \in L^\tau} x_{ij}^{\tau kl} d(i,j) \leq SFC_{dl}, \forall \tau \in \mathcal{T} \tag{10}$$

$$\sum_{(i,j) \in L} x_{ij}^{\tau kl} - \sum_{(h,i) \in L} x_{hi}^{\tau kl} = z_{ki}^\tau - z_{li}^\tau, \quad \forall i \in N, \forall (k,l) \in L^\tau, \forall \tau \in \mathcal{T} \tag{11}$$

Constraints 1 guarantee each VNF $k \in N_{vnf}^\tau$ required by the compositions $\tau \in \mathcal{T}$ is mapped to a N-PoPs $i \in N$ for the mapping to be accepted. Constraints 2 guarantee that each *end-point* $k \in N_{ter}^\tau$ required by the compositions $\tau \in \mathcal{T}$, is mapped to a N-PoP $i \in N$ in the viability region a_{ik}^τ for the mapping to be accepted. The auxiliary parameter $a_{ik}^\tau \in \{0,1\}$ has the value 1 if only if the *end-point* $k$is in the same mapping region as the N-PoP $i \in N$, or 0 otherwise.

Constraints 3 guarantee that only one SFC composition is mapped among the alternatives provided. Constraints 4 guarantee the assignment of all instances demanded by the VNFs $k \in N_{vnf}^\tau$ on N-PoPs $i \in N$. In this case, the binary variable w_{mi} which indicates that an instance with the function m was created on the N-PoP $i \in N$.

Constraints 5 guarantee that each virtual node $k \in (N_{end}^\tau \cup N_{vnf}^\tau)$ is mapped on N-PoP $i \in N$ different. This imposition tries to reduce the number of points without service, in case of failures in the SN. Constraints 6 guarantee that the processing capacity of each N-PoP $i \in N$ is assured for each demand of VNF required by each composition $\tau \in \mathcal{T}$. Similarly, but dealing with storage capacity, the constraints 7 guarantee that the storage capacity of each N-PoP $i \in N$ is guaranteed to process each required VNF demand for each composition $\tau \in \mathcal{T}$. Constraints 8 guarantee that the capacity of each N-PoP $i \in N$ in accommodating different VMs is respected.

Considering the chaining constraints, the set of constraints 9 guarantee that the available bandwidth of physical links is not extrapolated to the mapping of each composition $\tau \in \mathcal{T}$. Constraints 10 guarantee that the maximum delay of SFC_{dl} allowed for each SFC is respected, the delay being calculated according to the use of SFC links. The set of constraints 11 guarantee the mapping of virtual connections, in order of composition $\tau \in \mathcal{T}$, mapped on the physical paths of links $(i,j) \in L$ of SN.

3) Formulation: For maximize the profit and resource sharing, the objective function is given by the difference between the revenue paid by the clients of the accepted requests (SFC^γ) , the costs with the number of virtualization licenses (η^m), and the spending with the lease of InPs links (ϵ_{ij}), as the following:

$$\textsc{Maximize:} \sum_{\tau \in \mathcal{T}} y^\tau SFC^\gamma - \sum_{i \in N} \sum_{m \in F} w_{mi} \eta^m - \sum_{\tau \in \mathcal{T}} \sum_{(k,l) \in L^\tau} \sum_{(i,j) \in L} x_{ij}^{\tau kl} \epsilon_{ij} \tag{12}$$

$$\text{s.t. (1) - (11)}$$
$$y^\tau, w_{mi}, z_{ik}^\tau, x_{ij}^{\tau kl} \in \{0,1\}$$

---

[3] Each virtual node in the same SFC must be assigned to a different physical node. This action reduces the impact of a node infrastructure failure [23].

[4] Each virtual link $(k,l) \in L^\tau$ can be mapped onto a physical path containing one or more links.

IV. PERFORMANCE EVALUATION

This Section describes our simulation environment and shows the performance evaluation of our approach for VNF-SPC problem. As mentioned in Section II, this is the first contribution to VNF chaining and placement whit composition selection. Thus, VNF-SPC cannot be compared with other state-of-the-art algorithms. However, for analyzing purposes, we use approaches that do not use composition selection, to establish comparison values.

The simulations were performed in a computer, Intel Core i5 3^{rd} with 8GB of DDR3 RAM, using the Ubuntu 14.04.2 LTS operating system. The tests simulator used was implemented in C++ and the models executed through the IBM ILOG CPLEX V12.6.3.

A. Experimental Workloads

Several factors are involved in the NFVs networks scenarios parameterization, as well as a significant variability of equipment that can be used. Abstracting this values variation, this work considers a simplified model, with some values based on real costs of Amazon[5] and others, were reproduced from the articles [24]–[26] (the instances are available at http://www.dcc.ufmg.br/∼smaa).

In the tests, the SN is composed by 50 nodes (N-PoPs), where their geographical coordinates being the same as of the [24], randomly generated in 50×50 grid by the GT-ITM tool. Each pair of vertices of the substrate is randomly connected with a probability of 0.1, and whether the connection exists are inserted round-trip links. The resources of CPU_i, ST_i e VM_i supported by N-PoPs and the resources the BW_{ij} SN links correspond to integers numbers uniformly distributed between [32 : 64], [960 : 1920], [8 : 10] e [25 : 50], respectively.

To generate the SFCs, in SFCC problem, similarly to [10] and [20], were used VNFs precedence constraints jointly with the objective function of bandwidth minimization. However, we opted a treatment without the splits generation in the chain of VNFs. This action reduces the number of different compositions combinations and producing a more elemental VNF-FG graphs family. The SFCs compositions may vary in Single Flow or Branch Flow SFC (Figure 4) with the functions sequenced in line, following [5] definition.

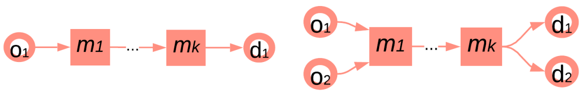

Fig. 4. An example of compositions Single Flow SFC and Branch Flow SFC.

A preprocessing was performed to generate the SFCs. Thus all permutations of valid compositions were made in line and analyzed to form the set $\mathcal{T}$, which contains the best SFC compositions to minimize bandwidth consumption. Each set $\mathcal{T}$ has at least 1 SFC, being limited to the maximum of 5 alternative SFCs, the size of that each set can vary depending on the number required VNFs and their precedence relations. The best SFC composition of the set $\mathcal{T}$ is called c^* and corresponds to the optimal composition in terms of minimizing bandwidth consumption for the client. The worst SFC composition within the set $\mathcal{T}$, i.e., with the highest bandwidth consumption, is called c^-. The VNFs characteristics required in each composition SFC correspond to uniformly distributed integer values and are presented in Table II.

TABLE II
PARAMETERS FOR VNFS GENERATION.

Functions	m_1	m_2	m_3
cpu_k^{τ} (vCPU)	[1 : 8]	[1 : 16]	[2 : 8]
st_k^{τ} (GB)	0	0	[50 : 100]
Unit cost ($/$t$)[a]	0.204	0.408	0.096
Flow implication bw[b]	+50%	+25%	maintain
Functions precedence[c]	(m_1, m_2)	(m_1, m_2)	(m_3, m_4)
	m_4	m_5	m_6
cpu_k^{τ} (vCPU)	[2 : 16]	1	1
st_k^{τ} (GB)	[50 : 100]	[100 : 300]	[300 : 400]
Unit cost ($/$t$)[a]	0.192	0.768	2.712
Flow implication bw[b]	−25%	−50%	−75%
Functions precedence[c]	(m_3, m_4)	(m_5, m_6)	(m_5, m_6)

[a] Values of cost SFC^{γ}, given in relation to the required VNFs and their lifetime.
[b] The required network function can increase, maintain or reduce the bw consumption.
[c] Functions (m_i, m_j) imply that m_i must precede m_j in the SFC.

The SFCs requests arrive at the provider according to a *Poisson* distribution with a rate of 40 SFCs per 1.000 time t for the larger instance (high density, 1000 SFCs), 10 SFCs per $1.000t$ for the medium-sized instance (medium density, 250 SFCs), and 5 SFCs per $1.000t$ for the smallest instance (low density, 125 SFCs). The performed tests consider $25.000t$ on total. Each SFC has a lifetime with exponential distribution of mean $\mu = 1.000t$. The following parameters correspond to evenly distributed integers. The VNFs number of an SFC and the number of end-poins are determined between [1 : 6] e [1 : 3], respectively. The geographical position (x_k, y_k) of each end-point is given between $x = [0 : 50]$ e $y = [0 : 50]$, and its mapping region by a radius between [10 : 16]. The maximum delay SFC_{dl} allowed for an SFC is 2.5 accounted for each VNF required. The initial band sent by each source is determined between [1 : 40].

The revenue SFC^{γ} is generated as in function of the duration time of each SFC and their required functions $m \in F$, the pricing is given by Table II. The other values used are the cost of each network instance for the TSP $\eta^m = \{204, 408, 96, 192, 768, 2.712\}$ and the cost of using the links $\epsilon_{ij} = 0.001$.

B. Evaluated Approaches

To do the analyses and comparisons of the VNF-SPC, the following approaches were proposed, through the ILP model:

- ILP-c*, performs the positioning and chaining of the SFC considering the optimal composition $\tau^* \in \mathcal{T}$, i.e., with lower total bandwidth consumption;

---

[5] Pricing of the *Amazon EC2 on demand*, https://aws.amazon.com/ec2/pricing/on-demand/,accessed09/12/2018

- ILP-c$^-$, performs the positioning and chaining of the SFC considering the worst composition $\tau^* \in \mathcal{T}$, i.e., with higher total bandwidth consumption;
- ILP-$\mathcal{T}$, performs the **selection**, positioning, and chaining of the composition $\tau \in \mathcal{T}$ which best fit the residual SN at the time of the mapping, i.e., analyzes all compositions.

C. Evaluation Metrics

The metrics used in this work are classic in networks virtualization and some used in [24].

Profit: Measured profit according to Equation 12.

Acceptance rate of SFCs: Is given by the ratio of the number of SFCs accepted concerning those that arrived until the considered time t.

Average link utilization: The utilization ratios for links are important in detecting possible bottlenecks and fragmentation of SN, which potentially lead to an increase in the number of rejections. The average link utilization rate is given by the sum of the ratio between bandwidth residual and total in time t of all the physical links.

Average node utilization: The utilization ratio for nodes can be separated into CPU and Storage. They are essential for information about the depletion of physical node resources, which potentially influence an increase in the rejections number. The average CPU utilization rate is given by the sum of the ratio between CPU residual and total in time t of all the physical nodes. Similarly, the average storage utilization rate is given by the sum of the ratio between storage residual and total in time t of all the physical nodes.

Execution Time: As the simulator was implemented in *C++*, was used the time.h to measure execution time.

D. Experimental Results

In the VNF-SPC problem, for determining the solution with guarantees of optimality, is required mapping all SFCs together, as proposed in the offline scenario, or using reconfiguration strategies. On the other hand, given the complexity of the problem, an exact and offline treatment makes the execution time unfeasible in medium or large instances. In real scenarios, the TSPs usually deal with large workloads and unknown the arrival dynamics of SFC requests.

According to [27], for apply reconfigurations, the TSP need to interrupt the data streams of the already mapped SFCs, remove the active connections and reconfigure them. This action can grow the computation time, increase the cost and probably cause problems in the quality of service provided. In this study, we do not enter into this analysis, and we decided not to utilize before-mentioned action.

Figures 5, 6 and 7 present respectively the final acceptance/rejection rates, the TSP profits and the computational simulation time for the three approaches proposed in scenarios of different densities. In the S1 and S2 scenarios (Figures 5), the approach was effective in maintaining a high acceptance rate. On the other hand, the scenario S3 has a high arrival rate of SFCs and tends to become fragmented, which implies in a small increase in the rejection rate. An SN is fragmented when it has a set of cannot be used physical components, even with idle resources. In this case, the components adjacent to the fragmented region do not have the resources available to establish the routing/mapping of an SFC on the region.

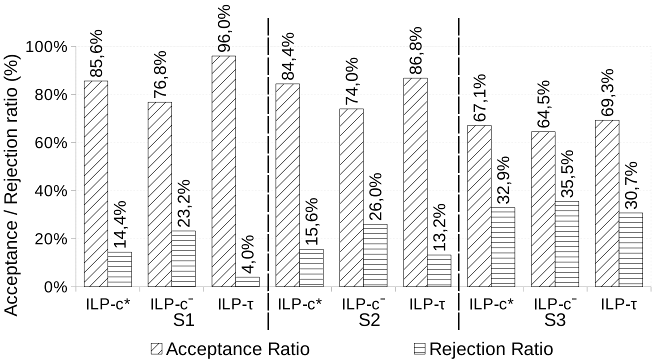

Fig. 5. Acceptance and rejection rates at the end of 25.000t, varying the density of the SFCs scenarios (S1, S2 and S3) and maintaining the same SN.

Analyzing the different approaches in Figure 5, the approach ILP-τ presented an acceptance rate higher than the other approaches, verifying the premise that an optimal SFC composition for the client may not be the most appropriate choice when mapped on residual SN in an online environment. Besides, the Figure 5 demonstrates that there is a difference between the ILP-c* and ILP-c$^-$ approach, where the choice of bad compositions can generate poor mappings, negatively affect the rate of acceptance and still to generate a decrease in profit (Figure 6).

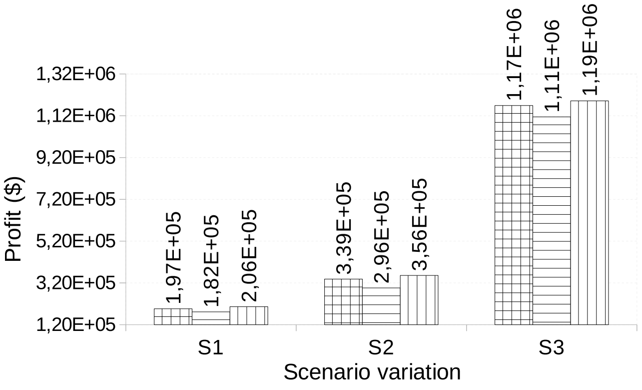

Fig. 6. Total profit at the end of 25.000t, varying the density of the SFCs scenarios (S1, S2 and S3) and maintaining the same SN.

The approach ILP-τ in addition to achieving higher acceptance rates (Figure 5), can generate an increase in TSP profit, as shown in Figure 6. This increase in profit occurs because alternative compositions generate flexibility in choosing the best composition for mapping in the residual SN, which leads to potential savings in the instantiation of VNFs and in a less costly linkage.

Although the ILP-τ approach is promising in terms of profit and acceptance rate, there is a cost of processing involved.

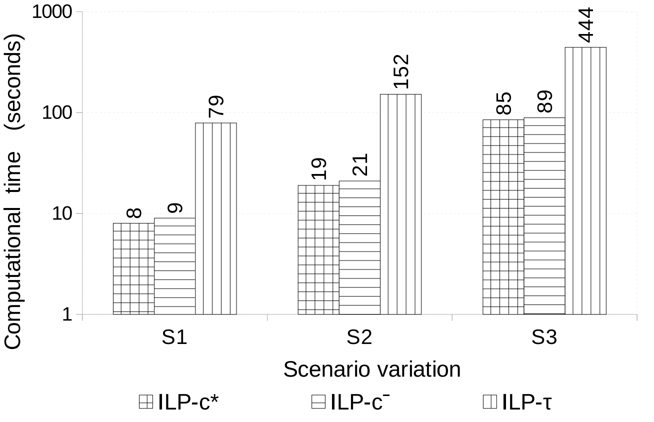

Fig. 7. Computational time for the process all SFCs, varying the density of the SFCs scenarios (S1, S2, and S3) and maintaining the same SN.

Figure 7 shows that the ILP-τ approach has a higher execution time compared to the other approaches. On the other hand, one aspect that should be considered is that although ILP-τ has a higher execution time, it does not necessarily need a preprocessing to solve the SFC composition in the optimally (which has exponential cost). The ILP-c$^-$ approach has the advantage of good execution time in the generation of SFCs and mapping since it considers SFCs viable even though a low-quality composition. However, ILP-c$^-$ has a lower profit and a higher rejection rate.

In the Figures 8, 9, 10 and 11, the stable use of bandwidth, CPU, and storage corroborate to the excellent behavior of the proposed model, where even in unfavorable and partially fragmented scenarios (S3) the fall in the acceptance rate was low whether compared to the large proposed load increase. It is possible to recognize the existence of the fragmentation based on the significant increase of the use of the links, expected consequence due to the abrupt increase of proposed load, and the low use of node resources (Figures 10 and 11).

Figures 5, 8, 9, 10 and 11 show that the approach ILP-τ has a higher acceptance rate, but this fact implies a slight increase in the use of links in relation to ILP-c$^-$. This fact occurs because when mapping more SFCs, ILP-τ tends to use more physical resources. Another point to be observed is that, although the ILP-τ approach has more significant use of links, this fact does not generate substantial impacts on the profit, since it is perceived that besides mapping more, the approach ILP-τ has a higher profit than the other approaches (Figure 6).

Albeit the ILP-c$^-$ approach theoretically has minimal bandwidth demand, it does not generate the highest profits. In this case, ILP-c$^-$ tends up performing worse positioning of instances in relation to ILP-τ, which results in higher instantiation costs, less sharing of functions and lower profits. Still, in Figures 8, 9, 10 and 11, it can be seen that even with high use of links, in an online scenario and without reconfiguration, the presented model presented an good performance, keeping the acceptance rate high and stable in most cases (Figure 5).

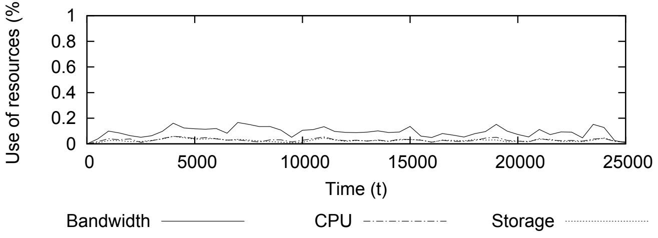

Fig. 8. Percentage of physical resources use: Scenario S1, approach ILP-c*

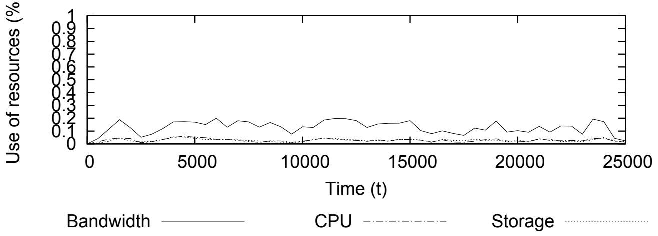

Fig. 9. Percentage of physical resources use: Scenario S1, approach ILP-τ

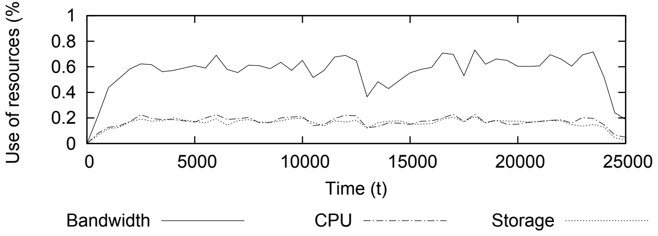

Fig. 10. Percentage of physical resources use: Scenario S3, approach ILP-c*

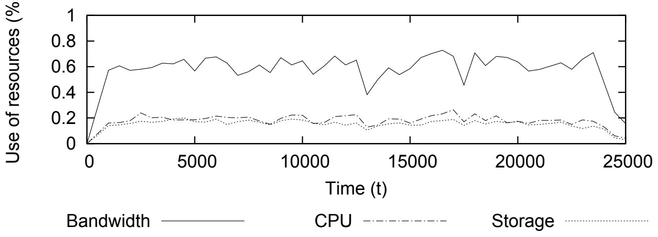

Fig. 11. Percentage of physical resources use: Scenario S3, approach ILP-τ

V. Conclusion and Future Works

To make Network Function Virtualization still further disseminated and evolved, efficient and practical new approaches are eminently necessary for the SFCs mapping, which tend to require more specific and complex functions. In this way, to generate more flexibility for the TSPs, this work proposed an ILP approach for the Positioning and Chaining of Virtual Functions of Network in online environments with Selection of Composition problem, called VNF-SFC.

The VNF-SFC proposes a flexibilization of the SFC compositions, seeking to increase the TSP profit; enhance the resources sharing; reduce the rejection rate, and to maintain a parsimonious use of the SN resources. Additionally, VNF-SPC aims to produce good solutions in environments requests low, medium and high-density.

It can be seen in the simulations, that the generation of the SFCs even when done with optimality guarantees, does not induce an optimal mapping on a residual SN. This fact occurs because the SFCs generation does not consider the residual state of the resources available at the moment of processing. It is noticeable that greater resource sharing tends to be beneficial by generating a lower cost of VNF instantiation, at this point, the approach ILP-τ differs from the other approaches and tends to profit more.

The online scenario proved to be challenging because the SFCs demands are only known at the moment of its arrival in the TSP, a fact that as perceived, hinders a better performance of the mapping algorithm. Despite the difficulty found in online environments, with a consequent slight fall in final acceptance and fragmentation in the high-density scenario, these situations are close to real aspects, and in practice, they are the ones that receive the most attention from the Scientific Community.

Given the combinatorial difficulty of the problem and the high observed resolution time, for future work, we intend to develop heuristics for the VNF-SPC problem. Another aspect related to future works is the analysis of the VNF-SPC in offline and periodic scenarios (with lot processing). We also pretend to explore more real and complex topologies to the SFCs and SN.

ACKNOWLEDGMENT

This work was partially financed by CNPq, FAPEMIG, and CAPES.

REFERENCES

[1] D. Bhamare, R. Jain, M. Samaka, and A. Erbad, “A Survey on Service Function Chaining,” *J. Netw. Comput. Appl.*, vol. 75, no. C, pp. 138–155, Nov. 2016. [Online]. Available: https://doi.org/10.1016/j.jnca.2016.09.001

[2] N. Herbaut, D. Negru, D. Dietrich, and P. Papadimitriou, “Service chain modeling and embedding for NFV-based content delivery,” in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–7.

[3] A. Laghrissi and T. Taleb, “A Survey on the Placement of Virtual Resources and Virtual Network Functions,” *IEEE Communications Surveys Tutorials*, pp. 1–1, 2018.

[4] L. Dinh-Xuan, M. Seufert, F. Wamser, P. Tran-Gia, C. Vassilakis, and A. Zafeiropoulos, “Performance Evaluation of Service Functions Chain Placement Algorithms in Edge Cloud,” in *2018 30th International Teletraffic Congress (ITC 30)*, vol. 01, Sep. 2018, pp. 227–235.

[5] 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)*, May 2015, pp. 98–106.

[6] 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, Sept 2016.

[7] 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, Feb 2015.

[8] R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. D. 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, Firstquarter 2016.

[9] X. Li and C. Qian, “A survey of network function placement,” in *2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC)*, Jan 2016, pp. 948–953.

[10] A. F. Ocampo, J. Gil-Herrera, P. H. Isolani, M. C. Neves, J. F. Botero, S. Latré, L. Zambenedetti, M. P. Barcellos, and L. P. Gaspary, “Optimal Service Function Chain Composition in Network Functions Virtualization,” in *Security of Networks and Services in an All-Connected World*, D. Tuncer, R. Koch, R. Badonnel, and B. Stiller, Eds. Cham: Springer International Publishing, 2017, pp. 62–76.

[11] J. Halpern and C. Pignataro, “Service Function Chaining (SFC) Architecture RFC7665,” 2015.

[12] B. Yi, X. Wang, K. Li, S. k. Das, and M. Huang, “A comprehensive survey of Network Function Virtualization,” *Computer Networks*, vol. 133, pp. 212 – 262, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128618300306

[13] 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, Fourth 2013.

[14] S. Lange, A. Grigorjew, T. Zinner, P. Tran-Gia, and M. Jarschel, “A Multi-objective Heuristic for the Optimization of Virtual Network Function Chain Placement,” in *2017 29th International Teletraffic Congress (ITC 29)*, vol. 1, Sep. 2017, pp. 152–160.

[15] 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)*, Nov 2015, pp. 50–56.

[16] J. Matias, J. Garay, N. Toledo, J. Unzilla, and E. Jacob, “Toward an SDN-enabled NFV architecture,” *IEEE Communications Magazine*, vol. 53, no. 4, pp. 187–193, April 2015.

[17] R. Cohen, L. Lewin-Eytan, J. S. Naor, and D. Raz, “Near optimal placement of virtual network functions,” in *2015 IEEE Conference on Computer Communications (INFOCOM)*, April 2015, pp. 1346–1354.

[18] M. Dieye, S. Ahvar, J. Sahoo, E. Ahvar, R. H. Glitho, H. Elbiaze, and N. Crespi, “CPVNF: Cost-efficient Proactive VNF Placement and Chaining for Value-Added Services in Content Delivery Networks,” *CoRR*, vol. abs/1803.06025, 2018. [Online]. Available: http://arxiv.org/abs/1803.06025

[19] 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)*, Oct 2014, pp. 7–13.

[20] J. Gil-Herrera and J. F. Botero, “A scalable metaheuristic for service function chain composition,” in *2017 IEEE 9th Latin-American Conference on Communications (LATINCOM)*, Nov 2017, pp. 1–6.

[21] M. T. Beck and J. F. Botero, “Coordinated Allocation of Service Function Chains,” in *2015 IEEE Global Communications Conference (GLOBECOM)*, Dec 2015, pp. 1–6.

[22] A. Baumgartner, V. S. Reddy, and T. Bauschert, “Mobile core network virtualization: A model for combined virtual core network function placement and topology optimization,” in *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, April 2015, pp. 1–9.

[23] ETSI, “Network Functions Virtualisation (NFV); Infrastructure; Network Domain,” Industry Specification Group (ISG), GS NFV-INF 005 V1.1.1, 2014.

[24] N. M. M. K. Chowdhury, M. R. Rahman, and R. Boutaba, “Virtual Network Embedding with Coordinated Node and Link Mapping,” in *IEEE INFOCOM 2009*, April 2009, pp. 783–791.

[25] N. Ghrada, M. F. Zhani, and Y. El-khatib, “Price and Performance of Cloud-hosted Virtual Network Functions: Analysis and Future Challenges,” *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*, pp. 482–487, 2018.

[26] S. Arajo, F. S. H. Souza, and G. R. Mateus, “Abordagens Descentralizadas para o Mapeamento de Redes Virtuais em Ambientes Multidomnio,” *Anais do Simpsio Brasileiro de Redes de Computadores e Sistemas Distribudos (SBRC)*, vol. 36, 2018. [Online]. Available: http://ojs.sbc.org.br/index.php/sbrc/article/view/2441

[27] S. M. A. Arajo, D. L. Guidoni, F. S. H. de Souza, and G. R. Mateus, “Managing virtual network embedding through reconfiguration and expansion,” *SIMULATION*, 2018. [Online]. Available: https://doi.org/10.1177/0037549717754134