Computer Networks 185 (2021) 107726

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Survey paper

NFV Resource Allocation: a Systematic Review and Taxonomy of VNF Forwarding Graph Embedding

Frederico Schardong [a,*], Ingrid Nunes [b], Alberto Schaeffer-Filho [b]

[a] Instituto Federal do Rio Grande do Sul, Rolante, Brazil
[b] Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, Brazil

ARTICLE INFO

Keywords:
Network Functions Virtualisation
Virtualised Network Functions
VNF forwarding graph embedding

ABSTRACT

The emergence of Network Functions Virtualisation (NFV) is drastically reshaping the arrangement of network functions. Instead of being built on dedicated hardware (network appliances), network functions are now implemented as software components that run on top of general purpose hardware through virtualisation, namely virtualised network functions (VNFs). From this paradigm-shifting technology arise two problems: (i) how to place VNFs in an NFV-enabled network; and (ii) how to chain these VNFs. These problems are jointly referred to as the VNF forwarding graph embedding (VNF-FGE) problem. Having efficient solutions to the VNF-FGE problem is key to the success of NFV because placing and chaining VNFs automatically and efficiently reduces network and computing resources, thus reducing capital expenditure (CAPEX) and operating expenditure (OPEX). In this work, we systematically review the literature on the VNF-FGE problem. We present a novel taxonomy for the classification and study of proposed solutions to this problem. Research challenges that remain unaddressed are also discussed, providing recommendations for future work.

1. Introduction

Network Functions Virtualisation (NFV) [1] has been proposed as an alternative to traditional service provisioning in the telecommunications industry. NFV decouples software from hardware, thus enabling their independent evolution. Moreover, this brings further flexibility to the network because it allows virtualised network functions (VNFs) and their interconnections, which are represented as a graph – namely, a *VNF forwarding graph* (VNF-FG) [2] – to be deployed on demand, anywhere in the network topology. Therefore, NFV allows network operators to have fine-grained control over the network using resources that match the actual traffic [2].

Decisions must be made before the actual deployment of VNFs in the network infrastructure, which are associated with two key problems addressed by existing approaches in the context of NFV resource allocation. First, the *placement* problem [3] consists of the specification of where to deploy a VNF or a set of VNFs. Second, based on the VNF locations, the physical path through which VNFs will communicate must be specified, taking into account specific ordering requirements. For example, external traffic may have to go through a layer of two firewalls to discard unwanted protocols before it is sent to an IDS for further inspection, to then reach application-level servers. The problem of finding a physical path to connect VNFs is known as the *chaining* problem [4]. The placement and chaining problems together comprise the *VNF forwarding graph embedding* (VNF-FGE) problem [5], which receives as input a VNF-FG and deploys it in the physical network.

In order to ease the deployment of VNFs, substantial effort has been made with the proposal of solutions to address the placement [6,7] or chaining [8–10] problems, or both [11–13]. Such approaches rely on different techniques and use varying inputs. Therefore, understanding their similarities and differences is key to identify which approaches are suitable to particular situations. We thus present in this paper a comprehensive systematic literature review of existing methods that aim to solve the placement or the chaining problems or, more generally, the VNF-FGE problem. We used a systematic method to select covered research work, which focused on the main conferences in the context of computer networks and network management. Proposed solutions to the VNF-FGE problem are classified in two main groups, *offline* and *online*. Offline approaches solve the placement and/or chaining problems by considering the network topology and one or more VNF-FGs to be instantiated altogether, while online approaches also take into account previously instantiated VNF-FGs, thus coping with demands that arrive over time. We introduce proposed solutions within these two groups and provide an in-depth analysis of their facets (or characteristics), such as technique style and allocation objective. These and other facets comprise a novel taxonomy, which is used to analyse, classify

* Corresponding author.
E-mail addresses: frederico.schardong@rolante.ifrs.edu.br (F. Schardong), ingridnunes@inf.ufrgs.br (I. Nunes), alberto@inf.ufrgs.br (A. Schaeffer-Filho).

https://doi.org/10.1016/j.comnet.2020.107726
Received 13 April 2020; Received in revised form 14 July 2020; Accepted 30 November 2020
Available online 4 December 2020

and compare proposed solutions covered in this systematic review and, possibly, future approaches.

Our work complements existing surveys that focus on general aspects of NFV. Herrera and Botero [5] surveyed research efforts towards NFV resource allocation, but did not investigate the more general VNF-FGE problem in an extensive manner. Sousa et al. [14] examined NFV orchestration and enabling technologies, but did not discuss the VNF-FGE problem and its many facets with the same level of detail as we do. Li and Chen [15] provided a historical perspective over NFV applications, but did not focus on the problem of placement and chaining of VNFs in the infrastructure. Furthermore, our systematic review expands the discussion started in narrower surveys: (i) Li and Qian [16] specifically surveyed NFV orchestration frameworks and broadly explained their placement strategies; (ii) Bhamare et al. [17] focused on service function chaining of both SDN and NFV architectures, categorising and discussing approaches solely by their optimisation objective; and (iii) Mijumbi et al. [18] broadly described NFV and its research problems, briefly touching VNF-FGE. We, instead, focus specifically on the VNF-FGE problem, providing an *in-depth* and *systematic* analysis of approaches targeting this problem through our novel taxonomy. The key differences between previous surveys and ours are summarised in Table 1. Our systematic review and taxonomy are useful for both researchers who aim to propose future VNF-FGE approaches and practitioners who aim to select suitable existing solutions to their network environments.

Before analysing existing work, we provide a background on NFV orchestration in Section 2 and describe our paper selection method and review methodology in Section 3. VNF-FGE approaches are analysed orthogonally by facets in Section 4 and discussed in detail in Section 5. Section 6 points out open challenges that still need to be addressed in the context of VNF-FGE and, finally, Section 7 concludes our systematic review.

2. Background on NFV

To provide a better understanding of the VNF-FGE problem, we first introduce in Section 2.1 an overview of the standard architectural framework proposed for NFV and then discuss the challenges associated with the orchestration of VNFs in Section 2.2.

2.1. Overview of the NFV architectural framework

Network Functions Virtualisation (NFV) has received significant attention after the publication of a white paper authored by leading telecommunications service providers (TSPs) in 2012 [1]. Soon after, the European Telecommunications Standards Institute (ETSI) was appointed as the entity to host the Industry Specification Group for NFV (ETSI ISG NFV), which is responsible for developing NFV standards. The NFV architecture specified by the ETSI is composed of two main components [2,19]: (i)*Virtualised Network Functions* (VNFs); and (ii) *Management and Orchestration* (MANO).

The core of NFV is a set of deployed VNFs, which are functional blocks that belong to a network infrastructure and have a well-defined behaviour and external interfaces. A VNF is the virtualisation of a network function (NF), which can be, *e.g.*, a dynamic host configuration protocol (DHCP) server or a firewall. The functional behaviour of an NF is usually independent of whether it is virtualised or not.

NFs are combined to form higher-level functions that represent end-to-end services, or network services. According to ETSI, “a network service (NS) is a forwarding graph of NFs interconnected supported by the network infrastructure” [2]. NSs have nodes connected by virtual links and, in addition to NFs, can include nested NF forwarding graphs. We illustrate an end-to-end NS in Fig. 1, provided by VNFs, which includes a nested forwarding graph. A VNF forwarding graph (VNF-FG) is a forwarding graph composed of virtualised NFs. Furthermore, VNF-FG and service function chain (SFC), which is an ordered set of service functions that compose an end-to-end service, are interchangeably used in the NFV literature. This term was borrowed from the context of software-defined networking (SDN). More recently, NFV and SDN have been pointed out as the foundations of network slicing [20], a verticalisation technology that allows multiple slices (end-to-end logical networks) to run on a shared physical network.

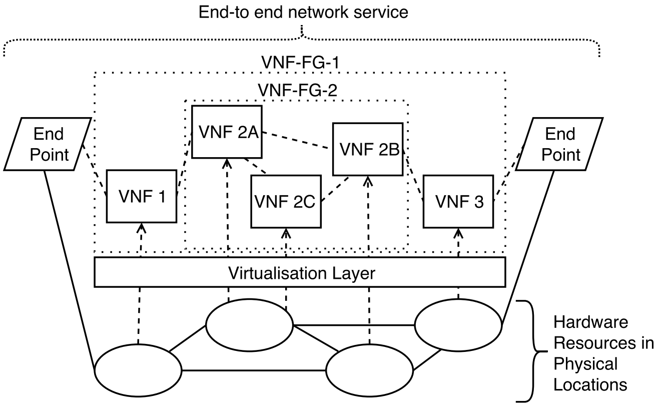

Fig. 1. Forwarding graphs and the supporting infrastructure [2]. In this example, a network service is served by two VNF-FGs, where the VNF-FG-2 is nested into the VNF-FG-1. Ellipses denote NFVI-PoPs, bold lines represent physical links, dotted lines represent virtual links and the mapping between VNFs and NFVI-PoPs is depicted through vertical dashed arrows.

2.2. NFV orchestration

Although the functional behaviour in NFV is provided by the VNFs, the MANO component is crucial because it orchestrates them so as to form NSs to achieve the goals of network operators. The orchestration of VNFs essentially involves three complementary decisions, which lead to three problems to be addressed, namely *VNF chain composition* (VNF-CC), *VNF forwarding graph embedding* (VNF-FGE) and *VNF scheduling* (VNF-SCH). The first, VNF-CC or, alternatively, selection problem [13,21,22], consists of the selection of a set of VNFs to collectively achieve a particular operator’s goal. As result, we obtain a VNF-FG.

The VNF-FG is a conceptual solution to achieve the operator’s goal, which must be embedded into the physical network. This requires deciding where to place and how to physically chain the VNFs, which are the two decisions associated with the VNF-FGE. Therefore, a VNF-FGE solution consists of a mapping of VNFs to NFVI-PoPs (placement solution) and appropriate paths in the physical network to connect each pair of VNFs (chaining solution). The VNF-FG solution must guarantee that the amount of available resources of NFVI-PoPs and links satisfies the requirements of the VNF-FG to be instantiated and its VNFs. Fig. 2 shows an example in which two VNF-FGs are embedded into an NFV-enabled infrastructure.

The VNF-FGE problem is similar to the virtual network embedding (VNE) problem, which is the primary resource allocation challenge in the area of network virtualisation [23]. In VNE, virtual routers are mapped to physical nodes, while virtual links that connect virtual routers are mapped to physical links [24]. Although both the VNE and VNF-FGE problems aim at finding the efficient allocation of virtualised components, there are two significant differences between them [11,22]. First, VNFs in a VNF-FG might have distinct functionality, *i.e.* they can be distinct NFs, while nodes in a VNE perform equivalent functions, *i.e.* routing. Second, VNFs in a VNF-FG have specific ordering. The requirement of supporting specific ordering between VNFs adds additional difficulty when finding adequate solutions.

Finally, the third problem to be addressed by an NFV orchestrator is the VNF-SCH problem [25]. The objective of this problem is to share VNFs among multiple VNF-FGs. It is built on the idea that VNFs are not

Table 1
Comparison with other surveys in the literature.

	Survey type	Proposal of a taxonomy	Focus on VNF-FGE	Covered VNF-FGE papers
Herrera and Botero [5]	Survey	Yes	Yes	2
Sousa et al. [14]	Survey	Yes	No	–
Li and Qian [16]	Survey	No	No	–
Li and Chen [15]	Survey	No	Yes	9
Bhamare et al. [17]	Survey	No	Yes	20
Mijumbi et al. [18]	Survey	No	Yes	21
This work	Systematic review	Yes	Yes	72

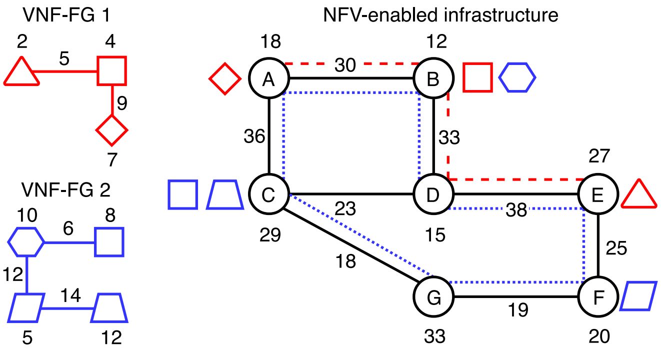

Fig. 2. Virtual network function forwarding graph embedding example. On the left-hand side, there are two VNF-FGs that have VNF and virtual link requirements to be fulfilled by the VNF-FGE solution. Each geometric shape represents a different VNF. The physical network and its available resources are located on the right-hand side. The VNF-FGE solution is illustrated by the geometric shapes located beside physical nodes (NFVI-PoPs), indicating where each VNF is deployed, together with the dashed and dotted links, which highlight how virtual links are instantiated in the physical links.

used all the time, and therefore can process packets of other VNF-FGs during idle periods of time. Ultimately, assigning tasks to idle VNFs decreases the number of deployed VNFs.

These three introduced problems, namely VNF-CC, VNF-FGE and VNF-SCH, are collectively referred to as the *NFV resource allocation* (NFV-RA) problem [5]. All these problems must be addressed by the MANO component that deals with the orchestration and lifecycle management of VNFs. In this work, we focus our attention towards the VNF-FGE problem, systematically investigating and analysing existing solutions.

3. Methodology

We conducted a systematic literature review to survey [26] the existing work in the area of VNF forwarding graph embedding. This means we followed a systematic method, which is explained in this section, to select the research work covered in our review. Further, this section also describes how we analysed the existing literature.

3.1. Selection method

Our literature review aims at covering research work that addresses at least one of the introduced decisions – placement and chaining – associated with the VNF-FGE problem. Therefore, we searched for research papers that contain two terms in their abstract: *virtual network function* and *placement* or *chaining*. Following the procedure of systematic literature reviews, we considered synonyms of these terms using as reference work published in the field [27–29]. Based on these terms and their synonyms, we formed our search string, presented below, that was used to query digital databases. Retrieved papers must have in their abstract, title or keywords at least one occurrence of the possible variations of the selected terms.

Table 2
Target conferences.

Acronym	Full name
INFOCOM	IEEE International Conference on Computer Communications
GLOBECOM	IEEE Global Communications Conference
ICC	IEEE International Conference on Communications
CNSM	IFIP International Conference on Network and Service Management
IM	IFIP/IEEE International Symposium on Integrated Network Management
NOMS	IEEE/IFIP Network Operations and Management Symposium
SIGCOMM	ACM Special Interest Group on Data Communication
CoNEXT	ACM Conference on emerging Networking Experiments and Technologies
HotNets	ACM Hot Topics in Networks
SOSR	ACM Symposium on SDN Research

(virtual network function **OR** VNF **OR** network functions virtualisation **OR** network function virtualization **OR** NFV)
AND
(placement **OR** place **OR** chaining **OR** chain)

As mentioned in the previous section, NFV received significant attention after 2012. Our review thus included papers published in the last seven years (2013–2019). Our goal is to introduce and discuss the existing literature, making an in-depth analysis of proposed solutions, comparing their inner-workings. Therefore, to make our work relatively concise, we limited the scope of our search. We surveyed only papers that appeared on flagship networking conferences and also on conferences focused on network management (the specific conferences we considered are shown in Table 2). We targeted conferences (rather than journals) because they have a shorter publication timespan, and because journal papers are typically predated by a conference paper and are therefore potently already covered by our review.

Results retrieved by searching the database may include papers out of the scope of this review. We thus specified inclusion criteria (IC) and exclusion criteria (EC), stated in Table 3, to determine whether a paper should be selected to be reviewed—papers should meet all IC and no EC. The IC state the scope of this literature review. Papers must not only have addressed the VNF-FGE problem, but also be general in terms of the kinds of supported VNFs and the number of supported NFVI-PoPs. Further, we excluded from the survey papers that were not published as full research papers, given that short and demo papers, as well as posters, typically represent early or ongoing work.

We queried the IEEE Xplore database on March 08, 2020, and the ACM Digital Library on June 30, 2020, and retrieved, respectively, 788 and 119 papers. By applying our IC and EC, we obtained as result 72 papers, which are those included in this literature review. We summarise the number of papers that did not meet our criteria in Table 4. To provide an overview of the papers covered by our survey, we summarise them in Fig. 3. This figure provides evidence of the increased interest in the topic of this survey in the last few years. We also distinguish papers by the type of solution they provide (offline or online), which is a key characteristic that distinguishes solutions to the VNF-FGE problem. This and other characteristics are used to classify and discuss published research work, as detailed in the next section.

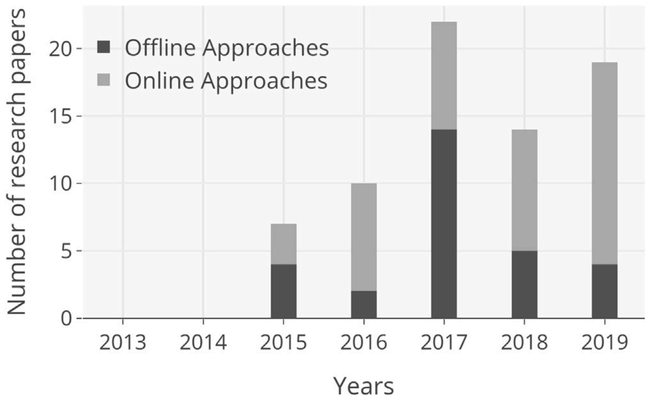

Fig. 3. Papers covered by our systematic review by year. No paper published in 2013 or 2014 satisfied our IC and EC.

Table 3
Inclusion and exclusion criteria.

Inclusion criteria	
IC-1	The paper includes the proposal of an approach that addresses least one of the decisions of the VNF-FGE problem.
IC-2	The approach proposed in the paper supports multiple NFVI-PoPs.
IC-3	The approach proposed in the paper is a technology-independent solution.
IC-4	The approach proposed in the paper is a context-independent solution.
Exclusion criteria	
EC 1	The paper is not written in English.
EC 2	The article is not a full or technical paper, *i.e.* it is a workshop, short paper, poster, demo, tutorial or etc.
EC 3	The research work does not have an empirical simulation or numerical evaluation.

Table 4
Selection of studies.

Results	Total
IEEE Xplore Digital Library	788
ACM Digital Library	119
Selected Conferences	217
Excluded by EC-1	0
Excluded by EC-2	44
Excluded by EC-3	0
Candidate Papers	173
Not Satisfied IC-1	70
Not Satisfied IC-2	5
Not Satisfied IC-3	22
Not Satisfied IC-4	4
Selected papers	**72**

3.2. Limitations

A systematic literature review follows a protocol to identify relevant research work. The key advantage is to reduce the researcher bias while conducting the study. NFV have been largely investigated in academia, and this led to many published papers on this topic. The inspection, classification and detailing of all published papers – without restricting our review to a particular set of conferences – would result in a tedious report for the reader to follow. Therefore, we contemplated only works published on flagship networking conferences and on conferences focused on network management, aiming to capture the most relevant works published in the fast-paced literature of NFV. This decision limits the extension of our manuscript by not considering works published in other conferences [30,31] as well as in journals [32,33]. Notwithstanding, works covered by our survey have been extended and published in journal venues [34–36]. These, however, were not included in our systematic review because they would confront our methodology. Systematically surveying journal publications will be subject of a new systematic review.

3.3. Analysis method and taxonomy

Each work investigated in our review was analysed from different perspectives, leading us to identify key characteristics that distinguish them. These characteristics are organised into a proposed taxonomy, which has five facets, as shown in Fig. 4, and is used to guide our discussion of the existing literature in later sections. Each facet is described as follows.

The first facet, *behaviour*, separates approaches into two groups, as highlighted in Fig. 3. *Offline* approaches consider one or more VNF-FGs and a set of NFVI-PoPs connected by link, both known *a priori*, and provide a solution of how the VNF-FGs should be embedded into the infrastructure. *Online* approaches also take into consideration a previous state of the network infrastructure, in which other VNF-FGs have possibly already been deployed. Consequently, online approaches can be used to gradually change the network infrastructure, with requests to instantiate VNF-FGs being received on-the-fly.

The second facet, *allocation task*, concerns the decision associated with the VNF-FGE problem that is addressed by the approach. As explained, this problem involves deciding where to deploy VNFs (*placement* task) and how to physically link deployed VNFs (*chaining* task). Approaches can be also classified as *both*, when they address both tasks.

Approaches may pursue a solution that achieves a particular *objective*, our third facet. For example, a particular place to deploy a VNF may lead to reduced network traffic but at the same time consume more resources. Therefore, the objective of the approach would indicate whether deploying this VNF in this location is a suitable solution. We identified four general groups of objectives: (i) *reduce network resources*, indicating the aim of reducing aspects related to the network traffic, such as bandwidth, latency or hop count; (ii) *reduce host resources*, indicating the aim of reducing computational resources of VNFs and NFVI-PoPs, such as processing delay, consumed CPU, memory and storage; (iii) *trade-off between network and host resources*, which considers both network and host resources and their trade-off; and (iv) *reduce economic cost*, focusing on increasing profit and/or reducing monetary costs.

To achieve the above objectives, approaches to address the VNF-FGE problem consider different *factors*. These can be characteristics of the network (*e.g.*, link latency and capacity) or of VNFs and NFVI-PoPs (*e.g.*, the number of CPU cores or RAM and storage amounts). Such physical and virtual factors are used as input by the proposed approaches to suggest an appropriate solution. Fig. 4 presents a non-exhaustive list of factors used by existing approaches.

Finally, the *technique style* facet specifies the nature of the approach proposed to accomplish at least one of the allocation tasks. Similarly to previous work [37–39], we classify them as: (i) *exact*, when the optimal embedding of VNFs and their connections are guaranteed; (ii) *heuristic*, when optimal embedding is exchanged for faster execution time; and (iii) *meta-heuristic*, when stochastic behaviour is incorporated to search the problem space thoroughly in reasonably practical time [40]. For each of these technique styles, we also provide a non-exhaustive list of alternatives.

In the next section, we proceed to discuss and present the categorisation of the investigated literature based on the facets of our proposed taxonomy. Approaches are then discussed in-depth in Section 5.

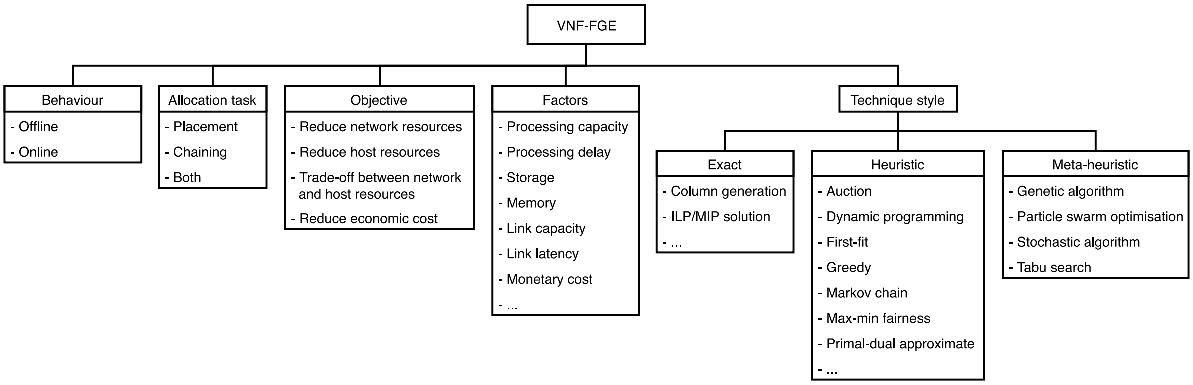

Fig. 4. Taxonomy used to classify the VNF-FGE approaches.

Table 5
Classification of reviewed articles using our taxonomy facets: (i) allocation task, second left-to-right column; (ii) objective, last row where N stands for *reduce network resources*, H for *reduce host resources*, NH for *trade-off between network and host resources* and E for *reduce economic cost*; (iii) behaviour, left-most column; and (iv) technique style, first row.

		Exact				Heuristic				Meta-heuristic			
Offline	Placement						[41–44]	[6]		[7, 45, 46]			
	Chaining					[8,47]							
	Both	[48, 49]				[22,50,51]	[11,52,53]	[54–58]	[59–61]	[62]			
Online	Placement		[63]				[64–66]		[67]		[68]		
	Chaining					[9,69–71]		[72]		[10, 73]			
	Both	[74]		[75, 76]		[28,77–80]	[13,81,82]	[27,29,83–91]	[12,92–97]			[98, 99]	
		N	H	NH	E	N	H	NH	E	N	H	NH	E

4. Analysis of facets

The previous section presented the methodology applied for selecting papers and our proposed taxonomy – composed of five facets – to classify these efforts in the context of the VNF-FGE problem. In this section, first, the research efforts are explicitly classified according to our facets as shown in Table 5. Second, we orthogonally discuss the literature by facet. Given that research work is split according to our first facet (behaviour) in the next section, we discuss here our four other facets.

4.1. Allocation task

Both the *placement* and *chaining* problems are NP-hard problems [100]. Nevertheless, the vast majority of the reviewed works concentrate on solving *both* the placement and chaining problems together, namely the VNF-FGE problem.

4.2. Objectives

Because VNF-FGE is an allocation problem, a solution is typically designed to achieve an optimisation goal, which can focus on a single or multiple aspects. For instance, in a scenario where strict SLAs about the time for serving requests are applied, a solution might allocate VNFs and their virtual connections to use the physical links with the lowest latency and VNFs with the lowest processing delay. We classify the allocation *objective* of the reviewed works as one of the four options: (i) *reduce network resources*; (ii) *reduce host resources*; (iii) *trade-off between network and host resources*; and (iv) *reduce economic cost*.

It is important to notice that these four objectives are related. For example, an algorithm that aims to improve network and host resources gives as output solutions with low network and host usage, which consequently implies in lower economic costs. However, for an objective to be classified as economic, it must consider economic factors explicitly. There are works that formalise network cost or embedding cost but do not translate those into monetary cost [54]. Moreover, there are research efforts that introduce ideas such as computing and network costs, which are defined as, for example, the number of CPU cores and amount of bandwidth, or the ratio between free and total CPUs as well as the ratio between free and total bandwidth of a link, and optimise their solution to minimise these costs [27,88,99]. In such cases, the allocation objective is not considered to *reduce economic cost*. Only approaches that specify a pricing policy [60], introduce revenue and payment formalisations [92,93] or use monetary values as cost units [59,61,67] and are optimised to reduce cost or to increase revenue are classified as to *reduce economic cost*. Finally, we point out that these objectives are not exhaustive and future solutions might therefore use innovative and currently unforeseen objectives not accounted in this taxonomy.

4.3. Factors

When designing an algorithm to tackle the VNF-FGE problem, one has to choose which characteristics will be modelled as part of the problem and, consequently, used in the proposed approach. Selecting common *factors* such as *processing capacity*, commonly formalised as the number of CPU cores, and *link capacity*, which is usually modelled as the bandwidth of a link, is arguably enough to develop a solution to VNF-FGE [50,79]. Solutions might include factors such as *processing delay* of VNFs, *storage* and *memory* of VNFs and NFVI-PoPs (or merely the amount of VNFs supported by NFVI-PoPs) as well as *link latency* and the number of hops of physical/virtual links to provide more applicable and realistic solutions. Moreover, the factors inherited from the physical and virtual networks are often balanced against the cost

Table 6
Factors.

	Offline	Online
Processing capacity	[11,22,41–61]	[10,12,13,27–29,63–68,72,73,75,76,78–82,84–89,91,92,94–98]
Processing delay	[7,11,47,49,59]	[10,66,68,78,84,88,90,95]
Storage	[6,22,47,53,54,60–62]	[10,27,29,67,74,77,82,88–90,92,93]
Memory	[60]	[13,66,67,76,82,88,89,92]
Link capacity	[8,11,22,41,42,48–62]	[9,10,12,27–29,64,67–69,71,73–79,81,82,84–98]
Link latency	[8,11,45–47,59,60]	[9,28,49,63,68–70,73,76,78,80–82,84–86,88,89,92,95]
Monetary cost	[44,45,47,60,61]	[12,67,92,94–97]

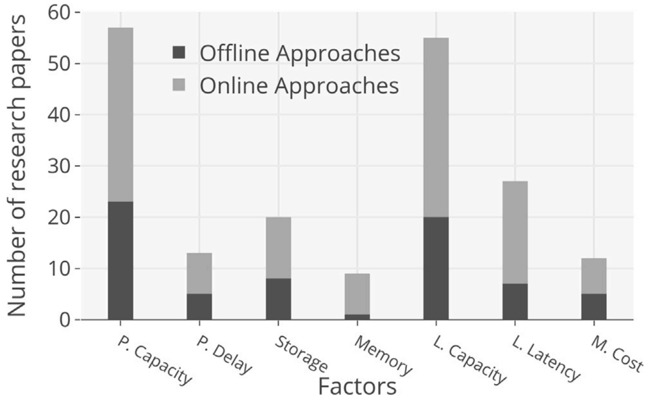

Fig. 5. A visual representation of the popularity of factors.

of using them. For example, the *monetary cost* of allocating a CPU in a public NFVI-PoP [67] or the monetary cost of using a specific physical link [61].

The most common factors found in the reviewed work are grouped in Table 6 and Fig. 5. The minimum and maximum amount of factors considered by the reviewed literature is, respectively, two [9,51,79] and ten [60]. The most common amount of factors (mode) is three. We refer the interested reader elsewhere[1] for a table that details the factors used in the reviewed articles individually.

4.4. Technique style

An algorithm that performs an allocation task considers different factors and aims to achieve an objective using an *exact*, *heuristic* or *meta-heuristic* technique. Exact algorithms for VNF-FGE and its subproblems find optimal solutions typically in non-polynomial time, given the NP-hardness of these allocation tasks. One of the exact algorithms reviewed provides polynomial-time solution for tree topologies [77].

Heuristics and meta-heuristics trade optimality for faster processing time. The vast majority of the reviewed work models the VNF-FGE problem as an integer linear programming (ILP) [7,48] or mixed-integer linear programming (MILP) [10]. Then, they propose a heuristic or meta-heuristic to efficiently solve it, while using the exact solution of the optimisation model for comparison [8,11,60,77]. In such cases, we only consider the heuristic or meta-heuristic contribution to classify the article. Efficient heuristic and meta-heuristic solutions outnumber inefficient exact solutions, as can be seen in Table 5.

In the next section, we proceed to the discussion of the various approaches that have been proposed to address the VNF-FGE problem. They are grouped according to our first introduced facet (behaviour) and we highlight key aspects of each approach considering the remaining facets.

5. State-of-the-art VNF-FG embedding

We now review the selected literature. Our discussion is split into offline and online approaches because they differ in nature. As such, they cannot be used interchangeably, even if they focus on the same allocation task, because offline approaches do not take into account VNF-FGs previously deployed in the network. First, we discuss offline approaches (Section 5.1) and then focus on those online (Section 5.2). Proposals that include both offline and online algorithms (*e.g.* [81]) are classified by their online contribution.

In both cases, a significant amount of the existing literature that addresses the VNF-FGE problem is based on linear programming [6,41]. Typically they use network configurations and constraints together with a selected objective function to specify an optimisation problem to be solved. Depending on the selected problem variables, the work may rely on, *e.g.* ILP or MILP. Available tools, such as CPLEX[2] or YALMIP [101], are then used to solve the specified optimisation problem, giving exact (or optimal) solutions. Because ILP is NP-Hard, exact solutions are commonly used as a baseline for the proposal of an alternative technique that may use a heuristic or meta-heuristic.

5.1. Offline

Twenty seven articles propose offline solutions to the VNF-FGE problem. They are grouped by allocation task and individually described below.

5.1.1. Placement

Five research efforts address the placement problem, proposing offline algorithms that give a solution using heuristics [6,41–44]. Although they focus on the same problem, they aim to find a solution to the problem achieving distinct objectives. Cohen et al. [6] aim to find a solution that minimises the VNF *connection and setup costs*. To achieve this objective, a linear program is specified, modelling the storage requirement and setup cost of each VNF, as well as the available resources of NFVI-PoPs. This was the first work that considered the possibility of having an instantiated VNF shared among multiple VNF-FGs. The resolution of the specified linear program is achieved with two algorithms; one that does not consider the NFVI-PoP resource limitation (unconstrained placement) and another that considers this (constrained placement). Both algorithms rely on a heuristic that assigns VNFs, based on their amount of storage size (descending order), to NFVI-PoPs.

With a different goal, two works focus on minimising the *number of VNF instances* [41,42]. Chi et al. [41] proposed an alternative specification of a linear program to solve the placement problem. Their specification models VNFs according to their processing capacity. Chi et al. [41] also rely on a heuristic, which reduces the placement problem into the bin packing problem [102]. The idea is to allocate VNFs at good bins (as opposed to bad bins), which are NFVI-PoPs that have their fullness evaluated as greater than a certain threshold. The fullness of a bin is given by the number of flows that traverse the NFVI-PoP. Their work is limited to a single VNF type. Sang et al. [42], in turn, proposed a greedy heuristic that iterates over NFVI-PoPs with unprocessed requests and places the combination of VNFs in an NFVI-PoP to process the largest amount of unhandled flows.

[1] http://www.inf.ufrgs.br/prosoft/resources/2020/computer-networks-nfv-slr.

[2] https://www.ibm.com/products/ilog-cplex-optimization-studio.

Tomassilli et al. [43] created two heuristics and one exact solution for the placement problem. They consider that all demands are known a priori and aim to minimise *host-related resources* such as licence fees or energy efficiency. They introduced: (i) a dynamic programming greedy heuristic that selects NFVI-PoPs with the lowest cost to host VNFs in the pre-computed path; (ii) an approximation algorithm, which randomly selects constraints from the ILP formulation to guide for approximately optimal solutions; and (iii) an optimal dynamic programming technique for tree topologies that runs in polynomial time for the special case of all traffic going upstream or downstream in the tree.

Lastly, Sallam et al. [44] also developed two greedy heuristics that aim to *maximise the total amount of fully processed traffic* by selecting the best NFVI-PoPs to host VNFs. Moreover, they take into account the monetary cost of using NFVI-PoPs along with their processing capacity. The first heuristic is tailored for NFVI-PoPs with uniform costs. It assigns each VNF to the NFVI-PoP associated with the highest value of a given objective function. Second, an enumeration-based greedy heuristic [103] is introduced for the case of heterogeneous price. It groups NFVI-PoPs that satisfy the budget constraint by the amount of VNFs already installed, then deploys each VNF in an NFVI-PoP of the group that presents the largest gain regarding the objective function. Furthermore, assuming that the chaining problem is solved, two approximations for the VNF-SCH problem are provided.

Three works develop meta-heuristics [7,45,46] to address the placement problem having the allocation task of *reducing network resources*. Two meta-heuristics [7,45] aim to reduce the *end-to-end service delay* while guaranteeing a certain level of reliability [7] or availability [45]. Chantre et al. [7] formulated a nonlinear mixed-integer program where VNFs are characterised by their processing delay and reliability. The authors developed a meta-heuristic that uses particle swarm optimisation (PSO) [104] to find the best amount of redundant VNFs and their respective NFVI-PoPs. PSO is a search algorithm based on the interaction of particles that share information. Based on local and global information, they migrate towards a global optimum. The nonlinear mixed-integer programming formulation is used as the fitness function, guiding the search for good solutions. Similarly, Yala et al. [45] developed a genetic algorithm (GA) that allocates VNFs at either edge clouds or a central cloud. The former presents low latency but has scarce resource while the latter has high latency but plenty of computational resources. Link latency and instantiation costs are taken into account by the GA algorithm, which finds near-optimal solutions despite the objective function. Instead of using a LP to model the network, Manias et al. [46] opted for a machine learning algorithm—decision trees [105]—to realise the placement of VNFs with the objective of *reducing the delay between VNFs*. The authors argue that such an approach drastically reduces the computational effort required to orchestrate VNFs, as the learning phase is conducted offline on top of a near-optimal heuristic proposed elsewhere [106]. Their evaluation shows that their meta-heuristic outperforms the heuristic used to train the model in half of the scenarios analysed.

5.1.2. Chaining

Three proposals assume that the placement of VNFs has already been done and concentrate on the chaining problem [8,47,107]. They aim either to maximise *traffic throughput* – *i.e.*, chaining as many VNF-FG in the network as possible – while guaranteeing that end-to-end latency constraints are satisfied [8,47], or minimise *packet overhead* [107]. Each proposes a heuristic that takes into account the capacity of links [8,47,107] or their delays [8,47], together with VNF throughput [8,107] or the packet processing rate [47,107] and VNF processing delay [47].

Jiao et al. [8] first chain VNF-FG requests without taking the latency constraint into account. A VNF is chained by balancing two values of each VNF deployed in the network that can perform the required work: (i) the VNF available resources; and (ii) the capacity of the shortest path from the current NFVI-PoP to the NFVI-PoP where the VNF is deployed. Then, if the calculated path violates the end-to-end latency constraint of the VNF-FG, the selected VNF and its NFVI-PoP are discarded, and another candidate is evaluated. This process is repeated until a path that does not violate the end-to-end latency constraint is found. Using a different approach, Xu et al. [47] divide the VNF-FG requests into subrequests so that each subrequest has the same packet rate as the request with the lowest packet rate. These subrequests are chained from their source node to the destination node passing through the NFVI-PoP that has the lowest cost and satisfies the end-to-end delay constraint. This constraint is the total link latency plus the processing delay of the requested VNF, which is already deployed. Finally, Wang et al. [107] created a chaining heuristic that takes as input the placement of VNFs, and for each VNF in a VNF-FG request: (i) calculates the shortest paths from the current NFVI-PoP to the NFVI-PoPs that host the requested VNF; (ii) discards the shortest paths that do not have enough bandwidth capacity for the current request; and (iii) chains the request to the VNF whose shortest path has enough bandwidth.

5.1.3. Both

Most of the research effort provides a combined solution for both the placement and chaining problems, focusing on various objectives and adopting different technique styles. Three works introduce heuristics to address the VNF-FGE problem with the objective of minimising *bandwidth consumption* [22,50,51]. These heuristics use processing capacity of VNFs and NFVI-PoPs, as well as the bandwidth of links. In Beck and Botero's [22] heuristic, each VNF of a VNF-FG request carries the information to which VNFs it can be directly connected. This gives the freedom to swap the location of VNFs of a VNF-FG with loose or no dependency constraint. If a VNF cannot be chained, then another VNF that satisfies the dependency constraint is used. The heuristic tries to place a VNF in a physical node directly connected to the predecessor to minimise bandwidth utilisation. In case there is no possible location to place a VNF, the heuristic backtracks the allocation of the previous VNF and tries to install it elsewhere. Differently, Ye et al. [50] developed a grouping strategy. They evaluate which combinations of VNFs mapped to the same NFVI-PoP result in higher bandwidth savings. These VNF combinations and their directly connected VNFs are called critical subtopologies (CS) of a VNF-FG. First, CS are placed and chained into the physical network. Next, the remaining VNFs of the VNF-FG request are embedded into the infrastructure. Finally, Gupta et al. [51] introduced a formalisation in which each VNF-FG request has specific start and end points in the physical network. They place and chain VNFs using a two-step solution. First, a heuristic groups NFVI-PoP pairs whose links match the shortest path between the start and end points of multiple requests. The user inputs the number of groups used in the heuristic. Next, a column generation [108] algorithm places and chains requests using the NFVI-PoPs of the groups formed in the previous step.

Differently from the works detailed above that rely on heuristics to provide a solution to the VNF-FGE problem to minimise bandwidth consumption, two exact [48,49] and one meta-heuristic [62] algorithms have been proposed to address the same problem with the same goal. The exact algorithms [48,49] use the processing capacity of VNFs and NFVI-PoPs to decide the allocation of VNFs, while the meta-heuristic [62] only allows the allocation of a single VNF per NFVI-PoP. All these approaches also evaluate the bandwidth of virtual and physical links. Carpio et al. [62] proposed a GA algorithm that improves the fitness function of each iteration by creating generations of possible solutions where the traffic towards a VNF is divided into different flows, which are load balanced towards replicas of the target VNF. Splitting the traffic and using multiple VNFs increase the number of possible paths that a VNF-FG can use and consequently improves the allocation of virtualised resources. The idea behind fitness functions is to evaluate possible solutions for a given problem. They are used to rank different solutions generated by a GA, which borrows techniques such as mutation and crossover from nature [109]. Huin et al. [48] created an ILP formulation for the VNF-FGE problem with a number of variables that increases exponentially with the VNF-FG given as input. The column generation technique, also used in an aforementioned work [51], is used to find optimal solutions [108]. An interactive process begins by using a relaxation of the ILP to find the best solution for each request. Then, restrictions are generated and added to each solution, if they improve the value of the linear relaxation. The process stops when improving mappings cannot be found. Finally, Tomassilli et al. [49] introduced two column generation [108] methods that output the exact solution for the VNF-FGE problem considering two back-up models: (i) disjoint virtual link between VNFs, *i.e.* having a backup path connecting all VNFs of a given request that is disjoint from the primary path; and (ii) single link protection, where only the path between two VNFs is redundant. It interactively relaxes ILP constraints until the best solution is found, similar to Huin et al. [48].

An alternative to minimise bandwidth consumption is to reduce *host resource usage* when placing and chaining VNFs. This was the goal pursued by three works that proposed offline heuristics [11,52,53] that map processing capacity of VNFs and NFVI-PoPs, and bandwidth of virtual and physical links, using different kinds of input. Luizelli et al. [11] aimed to reduce the number of VNFs instances to serve VNF-FGs using link latency, processing delay and the number of available licences as input. First, their proposal finds the minimum number of VNFs that meet the demand and transforms the objective function of the ILP model into a constraint, resulting in a more bounded model. After discovering the minimum amount of VNFs, any solution that uses the lowest possible amount of VNFs is considered a feasible solution. Kong et al. [52] add redundant VNFs and paths to guarantee that embedded VNF-FGs present the desired level of availability. The adopted objective function is to minimise the use of computing resources taking into account the availability of VNFs and links. First, it finds the shortest path between the start and end points. If the path does not meet the availability constraint using a maximum number of VNF replicas, a disjoint backup path is selected. Then, the number and location of replicas are determined. Lastly, Chen et al. [53] developed a greedy heuristic that tries to allocate VNFs of a VNF-FG on the NFVI-PoP where the previous VNF of the same forwarding graph has been assigned. This heuristic attempts to deplete the available resources of an NFVI-PoP before recurring to its neighbours. The authors showed that such greedy approach has similar (slightly worse) performance when compared to the GA algorithm proposed by Carpio et al. [62]

Differently from the approaches previously discussed, which aim to reduce either network or host resources, we now discuss offline algorithms for VNF-FGE that specifically aim to *jointly minimise both*. There are four heuristics [54–57] and one exact solution [58] with this objective. They rely on processing and link capacities to formulate solutions to the problem. Additionally, the importance of these resources is formulated as user preferences by Song et al. [54]. They treated the placement of a VNF as a state in a hidden markov model (HMM) [110], and the chaining between the VNFs of a VNF-FG as the hidden transition of states. Although transitions cannot be observed directly, factors such as computing resources of NFVI-PoPs and bandwidth of links are used to find good state sequences, which correspond to a resource-efficient path. Similarly, markov chains were used in [56], where different states represent distinct placement and/or chaining of VNFs. The chain is constructed such that states converge to the desired objective, reaching near-optimal solutions in acceptable time. In the third proposed heuristic [55], the constraints and variables of the ILP formulation that grow exponentially with the number of VNF-FGs are replaced to obtain a relaxed linear program to embed VNFs. The heuristic drops constraints and variables to a polynomial amount by selecting the path with the largest bandwidth. Then, the new linear program is optimally solved in polynomial time. Considering that both links and nodes are not completely reliable, Zhong et al. [57] introduced an HMM that embeds VNF-FGs and creates backup VNFs and links for the VNF-FGs whose allocation do not have the desired level of reliability. In the HMM, each hidden state represents an NFVI-PoP that can allocate some VNF. Transition between two states represents the probability of allocating two VNFs on two NFVI-PoPs, and is calculated considering the resource usage of links and nodes. The most likely sequence of hidden states is then predicted using a Viterbi-based algorithm [111]. Viterbi algorithm is a method for finding the most likely sequence of states, where a multi-stage graph is created, and nodes represent possible states. Proposing an exact solution, Wang et al. [58] formulated the VNF-FGE problem as a linear programming problem without host and network constraints, and used Simplex [112] to output the optimal allocation of VNFs and virtual links. Then, an algorithm iterates over VNFs and selects the most appropriate NFVI-PoP that has enough resources to host them, as well as the most suitable links to the next NFVI-PoP.

Finally, four heuristics have been proposed with the aim of reducing *economic cost* [59–61]. They all take into consideration the cost of using NFVI-PoPs and physical links. However, each heuristic complements this information with other varying inputs: link latency [59,60], storage capacity [60,61], processing delay [59] or memory, licence cost, availability of VNFs and links, and user preferences over these [60]. Chen et al. [59] used a technique similar to one introduced above [54], where the placement and chaining of VNFs are given by the analysis of the transition of hidden states in an HMM. Selecting an NFVI-PoP to host a VNF is a state of the HMM, and traversing all VNFs of a VNF-FG is regarded as the transition between states, which are hidden. In Chen et al.'s heuristic, the cost of allocating VNFs in NFVI-PoPs and resource usage is observed in the transition of states. If the obtained path violates the latency constraint of the request, then for each NFVI-PoP in the path, alternative NFVI-PoPs are evaluated and used if they reduce the path latency. Vizarreta et al. [60] used a greedy heuristic to ensure QoS while embedding VNF-FGs. To embed a request, their heuristic first finds the shortest path between the physical endpoints of the VNF-FG that satisfies the QoS constraints. Next, the path is expanded to support all the VNFs of the VNF-FG. The best NFVI-PoPs to host VNFs are the ones that produce the minimum cost, which includes the cost of using more network resources and additional software licences. Lastly, the heuristic proposed by Tastevin et al. [61] is composed of three steps. First, it finds the minimum amount of NFVI-PoPs to host the VNFs of the requests. To do that, it ranks the VNF instantiation requirements and greedily assigns the maximum number of VNFs to the same NFVI-PoP until all VNFs have been assigned. Then, NFVI-PoPs are ranked by a centrality metric [113] that evaluates how many times an NFVI-PoP is in the shortest path between the start and end points of the VNF-FG requests. This measurement is used to rank NFVI-PoPs in decreasing order, and the number of NFVI-PoPs obtained in the first step defines how many NFVI-PoPs are selected in this step. Finally, the placement and chaining of VNFs in the selected NFVI-PoPs occurs through a Viterbi-inspired algorithm [111], where nodes represent possible locations to deploy VNFs and the weight of their connections is the shortest path between the two nodes. By reducing the number of NFVI-PoPs and physical paths allocated, the cost of the solution is also reduced.

5.2. Online

Offline approaches address an NP-hard [100] problem and so those online. However, developing an online algorithm implies the ability to handle scenarios where incoming demands are unknown at design time. Therefore, supporting dynamically arriving requests, which is the reality of telecom carriers, adds extra difficulty to the conception of a feasible online solution. Likewise the offline works detailed above, we group the online approaches by allocation task.

5.2.1. Placement

From the online approaches, six focus specifically on the placement problem [63–68]. Despite these works, one introduces an exact solution [63], four propose heuristics [64–67] and one a meta-heuristic [68]. The only exact solution [63] aims to place VNFs on a carrier cloud infrastructure that uses an edge cloud (*i.e.* a cloudlet) considering the optimal balance between them. This is achieved trading-off the minimisation of the maximum *utilisation of the cloudlet* and the minimisation of the *allocation of computing resources in the cloud*, which conflict to each other. A proposed MILP model considers several allocation objectives – processing capacity of VNFs and NFVI-PoPs, the latency of the physical link that connects the carrier cloud and the cloudlet, arrival rate of VNF-FGs, and virtualisation overhead of VNFs – as well as user-defined preferences over them.

Zhang et al. [67] created a placement heuristic that uses the processing capacity of VNFs, the types of VNFs that can be instantiated in an NFVI-PoP and the monetary cost of renting short and long-term NFVI-PoPs with different pricing and billing cycles following Amazon pricing policies.[3] Moreover, backup NFVI-PoPs and their costs are incorporated to account for inaccurate predictions. The online heuristic aims to minimise the overall *cost over time* by: (i) observing current demand for each type of VNF; (ii) deploying VNFs in backup NFVI-PoPs to absorb flows unserved; (iii) predicting future demands using an online learning algorithm [114]; and (iv) renting/renewing backup NFVI-PoPs based on predictions. Zhou et al. [64] formalised not only node capacity but also the probability of a VNF-FG reaching peak data rate, and used it to formulate the expected computational resource consumption of each request. Their algorithm aims to maximise *host resources* to host the largest amount of VNF-FGs, considering their possible peak data rate. They argue that VNFs from a VNF-FG should not be placed on the same NFVI-PoP, as it reduces the chance of successfully vertically scaling VNFs to attend peak traffic. Thus, the heuristic places VNFs by sequentially deciding which of the neighbours NFVI-PoPs will have the most resources available after hosting the current VNF. You and Li [65] created a max–min greedy heuristic for dynamically arriving requests that aims to minimise the *maximal load among the servers*. For the VNFs of each incoming request it: (i) sorts NFVI-PoPs by their capacities in ascending order; and (ii) assigns a VNF to the least used NFVI-PoP, that is, the first of the sorted list. The authors' evaluation shows that this simple heuristic is faster to compute than other heuristics and orders of magnitude faster compared to an exact algorithm. Lastly, Li et al. [66] introduced a placement heuristic with the objective of maximising *revenue*, where revenue is defined as the weighted sum of utilised physical resources. Their solution consists of two phases: (i) a greedy mapping of VNFs to NFVI-PoPs based on first come, first serve; and (ii) a delay-aware VNF remapping, which remaps all VNFs of unsuccessfully embedded requests by greedily remapping the ones with highest delay violation first.

Differently from the previous works that proposed heuristics, Sun et al. [68] created a meta-heuristic with the objective of minimising the *increment of the total delay when placing VNFs* that uses simulated annealing [115], which is a technique that slowly reduces the probability of accepting worse solutions as the search space is explored, to guide a greedy heuristic towards a good solution. Considering the paths of all non deployed VNF-FGs, their greedy heuristic places the VNFs with the largest data arrival rate first in the NFVI-PoP that results in minimum delay increment.

5.2.2. Chaining

A few other online approaches targeted solely on the chaining problem, providing either heuristics or meta-heuristics. The five research papers that introduce heuristics that concentrate on either improving different *network-related characteristics* [9,69–71] or *balancing network and host resources* [72]. They all model the latency and except for [70], consider bandwidth of virtual and physical links. Moreover, processing delay of VNFs [9,10] and processing capacity [10] are also used in particular models.

Thai et al. [9] proposed a two-step heuristic that aims to minimise the *end-to-end latency* of incoming requests. For each VNF of a VNF-FG request, it greedily selects VNFs already placed in the network to be chained, being the best candidate the VNF with the lowest path's latency with respect to the predecessor VNF. Then, VNFs of the VNF-FG that are allowed to be chained in a different order are randomly swapped. This step aims to reduce the path's latency and is repeated while improvements are made. Differently, Jia et al. [69] introduced a heuristic that chains VNFs aiming to maximise *network throughput*. Each physical link in the network is associated with a normalised weight that depends on its free bandwidth. The weight of the links in the shortest path between the source and destination NFVI-PoPs of a request are regarded as the cost of embedding it. If the total cost does not exceed a specific amount, then the request is accepted. This heuristic also evaluates links latency to guarantee that requests with end-to-end delay constraints are respected. Sallam et al. [70] introduced an exact algorithm and a heuristic for the chaining problem, along with an ILP formulation for the placement problem. Their heuristic receives a directed graph representing the NFVI-PoPs and their links and gives as output a valid path for the request by performing three steps. First, it creates a new directed graph where vertexes represent embedded VNFs, and edges represent all possible paths between them, these are annotated with the accumulated link delay of the underlying physical links, Then, it removes vertexes that only have outgoing edges, except for the source. Last, it submits the new graph to any shortest path algorithm such as Dijkstra's algorithm. The returned path is then used as the chaining between the already deployed VNFs to deal the VNF-FGE request. Gao and Rouskas [71] introduced an ILP formulation with a penalty function that indicates the congestion of a physical link. Then, they concluded that minimising the *maximum edge congestion*, which is their objective, is the same as finding the shortest valid walk [116]—finding a walk from a source node to a destination node that visits at least once each node of a given subset of nodes—on the graph that represents the physical network. Their online heuristic first chains a VNF-FG using a shortest path tour algorithm, then verifies if any selected physical link is more congested than a given value. If this holds, the process is repeated with a stricter constraint until no chaining is possible. The last solution found is then returned.

With a different goal, Huang et al. [72] developed an online chaining heuristic that not only decides the path to connect NFVI-PoPs, but also decides which NFVI-PoPs that host the requested VNFs are going to be used. For each VNF in a VNF-FG request it selects the least loaded VNF already installed in the network. Next, to connect them, the path that presents the lowest value despite the balance between minimising communication cost and stabilising traffic prediction queues is selected. The balance is performed by user-inputted preferences.

Two online approaches introduce meta-heuristics to tackle the chaining problem [10,73]. They also model the latency and bandwidth of virtual and physical links. Moreover, processing delay of VNFs and processing capacity are also used by Alameddine et al. [10]. These authors use a tabu search algorithm [117] to chain VNF-FGs dynamically with the objective of maximising the *number of accepted requests*. Tabu search is a meta-heuristic that accepts moves that decrease local solutions, which is useful if the search gets stuck in a local minimum, adding prohibitions to discourage returning to previous solutions. First, an initial assignment of VNFs from VNF-FGs to VNFs already deployed is made at random. In this work, VNFs can participate in multiple VNF-FGs. Next, their connections are made through the shortest path with enough bandwidth. Then, the tabu search algorithm performs a series of moves that swap selected VNFs to improve the initial solution and their shortest path are recalculated. Zhou et al. [73] created a meta-heuristic using deep learning to address the chaining problem. They

[3] https://aws.amazon.com/ec2/.

argued that deep learning can abstract and learn the details of complex data and therefore can be used in this problem space. With the objective of minimising *end-to-end delay*, their system works in two phases: offline and online. In the former the deep learning network is trained and in the latter it is used to decide in which NFVI-PoP the VNF-FG will be chained next, in a hop-by-hop manner, considering its starting and ending nodes. Then, ordering and resource constraints are verified and, if some is not satisfied, the request is rejected.

5.2.3. Both

Similarly to offline work, most of the online approaches address both the placement and chaining problems. Six research efforts have *network-related objectives* [28,74,77–80]. Except for one technique [80], they all assess link capacity, along with link latency [28,78,80], processing capacity [28,78–80] or the number of VNFs hosted in NFVI-PoPs [74,77].

With the goal of minimising *link with the maximum load*, Ma et al. [74] proposed an algorithm assuming that the order of VNFs in a VNF-FG is irrelevant. The algorithm sorts the VNFs by their traffic changing behaviour, *i.e.* VNFs either increase or decrease the traffic rate. Then VNFs are placed and chained by: (i) traversing the network from the start NFVI-PoP of the request towards the end NFVI-PoP selecting the path with the maximum bandwidth available and placing VNFs that decrease network traffic until all VNFs are placed; (ii) traversing the network from the end NFVI-PoP placing expanding VNFs; until (iii) a VNF visited in the first step is found. The path that connects the start and end NFVI-PoPs is the chaining path of the VNFs already placed.

In order to embed requests with the *least cost given by load of a link*, Ma et al. [77] developed three optimal algorithms with polynomial-time solutions for topologies with a unique path between any pair of nodes (*e.g.* a tree). Each algorithm handles a different type of embedding request, which can be: (i) requests with no ordering requirement between VNFs; (ii) requests with totally-ordered VNFs; and (iii) requests with partially-ordered VNFs. Then, for the case of topologies with multiple paths between any pair of nodes (*e.g.* a mesh), a greedy heuristic based on Dijkstra's algorithm is used to build a path from the start NFVI-PoP that has enough resources to host the VNF-FG. Once a path is found, then Dijkstra's algorithm is executed for the last time to find the path with the minimum cost from the current NFVI-PoP to the end NFVI-PoP. The obtained path is used as input to one of the optimal algorithms, depending on the type of VNF-FG to be embedded. Similarly, Liu et al. [78] proposed a framework that allocates VNFs of the same type from multiple VNF-FGs into one or more processing cores of the same NFVI-PoP. First, a dynamic programming algorithm is used to optimally join the requests. Second, either an optimal or an approximation algorithm can be used to allocate the graph from the former step into the infrastructure.

Focusing on minimising overall *bandwidth usage*, Jalalitabar et al. [79] developed a heuristic. It orders VNFs to be embedded by CPU requirement and splits them into groups of inter-dependent VNFs, which results in groups independent from each other. Then, the first VNF embedded of the first group is the one with the highest CPU requirement, while the following VNFs of the same and following groups are embedded in the closest NFVI-PoPs that have enough CPU to host them.

Also introducing heuristic, Li et al. [28] aim to maximise the *number of accepted requests*. The heuristic ensures their delay limits are met on fat-tree topologies. They added pods, racks and tenants to the data centre model. Requests from a tenant that use the same type of VNFs are grouped to reuse common VNFs. A relaxed ILP formulation solves the placement problem, and another relaxed ILP solves the chaining problem. Incoming requests are grouped with previous, such that already deployed VNFs are reused to obtain near-optimal results proactively.

Huang et al. [80] created a system where VNFs can be shared among multiple SFCs at the same time, if they have enough processing capacity. Moreover, VNFs can be scaled and/or migrated to new NFVI-PoPs if doing so allow current and new requests to be embedded. The decision making heuristic works by: (i) ordering VNFs considering their dependency to reduce VNF instances, then for each VNF; (ii) trying to reuse existing VNFs, and if it fails; and (iii) either scaling or migrating it to another NFVI-PoP. Both scaling and migration are driven by a metric based on resource usage that *balances processing capacity of nodes and accumulated link latency*.

Three approaches concentrate on the objective of minimising the *consumption of host resources* to handle both placement and chaining [13,81,82]. The first [13] evaluates CPU and memory of NFVI-PoPs and VNFs, along with provided user preferences over the allocation of these resources to minimise the *usage of CPU and memory of NFVI-PoPs*. To solve the VNF-FGE problem, autonomous software agents control NFVI-PoPs. The belief–desire–intention (BDI) reasoning architecture [118] enables these agents to make reactive and proactive actions towards the fulfilment of their intentions. Once an agent decides to achieve some objective, such as having a suspicious flow investigated, which can be motivated by network events or internal reasoning, it triggers an auction so that the decision of which VNF to be instantiated, its placement and chaining are collectively made. The agent that provides the lowest cost to activate a VNF wins the auction. Wen et al. [81] created a greedy algorithm that embeds requests aiming to minimise the *number of VNF instances* in the network. The approach evaluates the bandwidth and latency of virtual and physical links, but does not specify what resource from NFVI-PoPs is consumed by VNFs, which is only referred to as "resource". To avoid the resource consumption of instantiating new VNFs to handle new requests, the algorithm reconfigures underused VNFs to handle more requests through a three-step process. First, NFVI-PoPs are sorted in descending order of their available capacity. Second, all VNFs already deployed are organised by their type in lists, which are sorted in ascending order of the VNF requirement despite the NFVI-PoP resource. Third, NFVI-PoPs that have enough resources to host a group of VNFs of the same type receive a new VNF with their accumulated requirements, while the original VNFs are removed from their NFVI-PoPs. This step accounts for the bandwidth requirements of the VNF-FGs to which the reconfigured VNFs belong. Aklamanu et al. [82] included not only processing capacity but also memory and storage in their model of VNF and NFVI-PoP, which aims to minimise the *number of active NFVI-PoPs*. VNFs with high requirements are placed first at NFVI-PoPs with high usage, in order to saturate a server before using a new one. Next, chaining is performed by an extended A* path finding algorithm [119], where the cost function uses either the number of hops or a tie-breaker function that balances bandwidth and latency, in case of multiple paths have the same hop count.

Approaches optimised towards *both network and host factors* can be split into two groups: those that use cost measurements to place and chain VNF-FGs [29,75,76,83–87,98] and those that use alternative measurements [27,88–90,99]. Approaches that use cost measurements aim to minimise the costs. Different cost factors can be used to estimate an overall cost by combining the cost of individual factors [29,75,83–86]. Despite the cost factor, approaches in this group use processing capacity of NFVI-PoPs and VNFs and link capacity [29,75,84–86], together with NFVI-PoPs storage capacity [29].

Zhang et al. [83] tackled the VNF-FGE problem in multicast topologies, which have sets of source and destination NFVI-PoPs, where for each destination a VNF-FG embedded into the network must be traversed to process the traffic. They model a Markov Model where states represent distinct solutions to the problem returned from a heuristic based on previous work [120] and the transition between states is driven by cost changes. The heuristic connects the source NFVI-PoP to one of the destinations through the path with the least cost, which is as a metric defined by the operator, suggested to be the rate of *bandwidth* *utilisation for links and resource usage for NFVI-PoPs*. Using a distributed strategy, Feng et al. [84] introduced a solution where NFVI-PoPs rely on local knowledge about the embedded SFCs to minimise *network and host usage*. At every timeslot NFVI-PoPs: (i) compute a transmission utility measurement for itself and neighbours despite unserved requests; (ii) compute a processing utility for itself; (iii) compute optimal allocation based on these utilities; and (iv) assume the highest utility locally evaluated by NFVI-PoP is the global maximum. They argue that this technique outputs close-to-optimal solutions.

From a different perspective, Feng et al. [29] created an integer and a non-integer program to model the VNF-FGE problem, both aiming to minimise the *use of physical infrastructure*. They show that the non-integer version captures the capacity of splitting requests to improve resource utilisation. A linear-time heuristic proposed to solve the ILP model transforms it into a set of fractional knapsacks problems, one for each pair of NFVI-PoPs and their connecting link, considering the capacity of the NFVI-PoP and the link.

Feng et al. [75] modelled the embedding of VNF-FGs as an ILP-optimisation problem, where both cost and revenue are specified based on resource consumption and it aims to minimise *the mapping of virtual nodes and virtual links*. These metrics are used by an ILP solver to find optimal solutions. Xie et al. [85] introduced a MILP model that supports multicast VNF-FGs, *i.e.* requests with multiple sources and/or multiple destinations. To attack this NP-hard problem, they created a heuristic that: (i) adds a hypothetical source connected to all the sources from the multicast topology with unlimited bandwidth and 0 delay; (ii) calculates the shortest path to the closest NFVI-PoPs that can host the largest amount of VNFs; and (iii) computes the shortest path between the last deployed VNF and all destinations. Liu et al. [86] created a multi-stage heuristic that embeds VNF-FGs into the physical infrastructure that aims to minimise the *network and host costs*. Each stage represents the possible embedding locations of a VNF from the request. The transition between stages is done through a transition function that selects an NFVI-PoP to host the next VNF such as to minimise the network cost. Since only neighbouring NFVI-PoPs are evaluated by the transition function, it also solves the chaining problem.

Spinnewyn et al. [87] built on top of Beck and Botero's [22] heuristic. They expanded the latter by allowing network services with bidirectional chaining constraints and optional VNFs. To cope with the added complexity, two heuristics were introduced: a backtracking recursive heuristic derived from [22] and a Monte-Carlo tree search strategy [121]. Both heuristics rely the same core routine that finds unfinished VNF-FGs embedding and from the last allocated VNF a breadth first search procedure searches the closest NFVI-PoPs that have enough capacity to host the next VNF. Khezri et al. [98] conceived a meta-heuristic for attacking the VNF-FGE problem through Q-learning [122], which balances exploration and exploitation of different states by giving agents rewards that guide them towards the objective. In their system, the states represent the allocation of virtual resources and the reward is given by a function that penalises not achieving the desired reliability and choosing servers and/or links with not enough capacity for the current request. Neural network is used to guide agents in the search of feasible solutions. Considering that some VNFs could be shared by multiple requests, such as anti-virus, Mohamad and Hassanein [76] constructed an ILP model whose main contribution is having embedded VNFs being used by multiple VNF-FGs, *i.e.* sharing its unused resources.

Six other approaches aim to minimise the use of network and host resources [27,88–91,99], but do not rely on cost measurements. In addition to the evaluation of bandwidth of links and storage capacity of NFVI-PoPs, other factors such as NFVI-PoP and VNF processing capacity [27,88,89], their memory, link latency and processing delay are also used [88,89].

With the objective of reducing the *number of backup VNFs and overall physical resource consumption*, Fan et al. [88] introduce a model to measure the availability of a VNF-FG, which depends on how the VNFs are connected and their availability. This model is added to the VNF-FGE formalisation, and a greedy heuristic uses off-site NFVI-PoPs to provide redundancy, and thus improve availability. It first embeds a request to the data centre by calculating the shortest path between the ingress and egress NFVI-PoPs, and places VNFs along the way. Second, the two VNFs with the lowest availability are provided each with a backup in an off-site data centre. Kuo et al. [27] built their solution on top of the idea of reusing the spare processing capacity of VNFs to serve multiple requests. The concept of reuse factor (the number of VNFs from a VNF-FG request assigned to existing VNFs) is introduced. A mathematical model guides the dynamic programming algorithm towards feasible solutions that maximise the *reuse factor while minimising the path length*. This algorithm incrementally finds sub-paths that efficiently reuse VNFs and concatenates those sub-paths as the final solution.

The GA proposed by Rankothge et al. [99] allows dynamic changes in the physical infrastructure. They modelled the fitness function to maximise both *server and network usage*. It evaluates the amount of NFVI-PoPs used in a solution, the number of links used and how much these two factors changed from the previous state. User preferences allow the operator to change the importance of any component of this function. Varasteh et al. [89] created a heuristic that: (i) decides where to place a VNF based on a list of NFVI-PoPs that can be reached given a minimum amount of bandwidth, which is constructed in linear time; and (ii) chains the VNFs through a shortest path algorithm that balances power consumption and maximum link delay. Weighting parameters are used to selected the shortest path. Initially, high weight for low power consumption and low weight for link delay are used, which are gradually shifted until an acceptable solution is found. Fei et al. [90] created a heuristic that proactively deploys and maintains unused VNFs to avoid having to wait for the instantiation of VNF instances and potentially delaying or denying requests. An online convex optimisation algorithm called FTRL [123] is used to predict the amount and type of VNFs needed in the next time slot, based on current and past requests. Once new requests arrive, new VNFs are deployed, if needed, through a heuristic that selects NFVI-PoPs with the maximum residual space available. Next, a chaining heuristic that selects the paths with the largest amount of bandwidth available is employed to connect the VNFs. Lastly, Anwer et al. [91] introduced an inflation heuristic, where VNFs that output more traffic than that received as input are placed closer to the destination, while VNFs that reduce traffic are placed closer to the source of the VNF-FG. Having all VNFs from different VNF-FGs placed, their algorithm then decides what VNFs will be used for which VNF-FG and their connecting paths by computing the shortest weighted path. The weight for each path contemplates the physical network distance and inflation factor of VNFs.

Finally, seven online approaches tackle the VNF-FGE problem with the objective of maximising the *revenue of infrastructure providers* [12, 92–97]. While they all map link capacity, other factors such as link latency [92,95], hop count [12], NFVI-PoP and VNF processing [12,92, 94–97], memory [92], storage [92,93], power consumption [92] and reliability of VNFs and links [12] are also used.

The work of Rancheg et al. [92] aims to improve the service provider's profit by reducing energy consumption while considering different energy prices over NFVI-PoPs. They use ILP to model energy consumption of servers and add financial penalty paid to clients if a request's end-to-end latency is not satisfied. Three algorithms are proposed to solve the modelled VNF-FGE problem: (i) an exact solution that embraces SLA violations if it is profitable, *i.e.* revenue is higher than penalty and energy costs; (ii) an exact algorithm that does not accept SLA violations; and (iii) a heuristic that restricts the search space by only considering paths with resource usage less than a certain threshold. The resource usage of a path contemplates the use of NFVI-PoPs and links. Sun et al. [93] used a Fourier-Series-based prediction method [124] to anticipate the amount of VNFs that will be used in the next time frame. In their model, placing VNFs a priori have a lower cost than doing so in the time of need. This information is used as input to an algorithm that first releases the allocated VNFs and links of expired VNF-FGs and then embeds the predicted VNFs. The decision-making process that evaluates the best NFVI-PoP to host predicted VNFs considers their resource usage and the proximity (in hop count) to where the VNFs of the same type are currently deployed. Finally, once new requests arrive, a heuristic merges existing and incoming VNFs. Then, it places and chains them based on shortest path, using previously deployed VNFs. Soualah et al. [12] modelled physical links not only using their bandwidth capacity but also reliability, which is characterised by the mean time between failures. To solve the VNF-FGE problem, a decision tree is proposed where each node represents the placement of a VNF. The connection between a father and a child node translates to the mapping of the virtual path connecting these two VNFs. The tree is incrementally created for each incoming request with the help of Monte-Carlo tree search strategy [121], which also guides the decision of the best embedding. To increase the service provider profit, the algorithm rejects requests when there are many unreliable links as the service provider has to pay penalties for SLA violations.

Ma et al. [94] introduced an online heuristic that prioritises the allocation of incoming requests into idle VNFs, therefore avoiding instantiation costs. The embedding process combines shortest path with profit maximisation to decide where to instantiate VNFs and virtual paths. The system periodically: (i) migrates VNFs to less expensive data centres; and (ii) release idle VNFs, avoiding requests being denied because of unavailable resources. The former action is triggered after some pre-defined amount of time and the latter when a VNF has been idle for a while. Ma et al. [95] introduced a comprehensive ILP model of the VNF-FGE problem in its offline variant, providing a heuristic to solve it aiming to *maximise the total revenue* while meeting end-to-end delay SLCs. This heuristic tries to find a suitable path to connect the ingress and egress nodes of a SFC request using an approximation algorithm. Next, this heuristic is invoked by an admission control policy that dynamically evaluates whether accepting an incoming request results in high enough profit or not. It takes into account the current state of data-centres and the required resources of a given request. Chen et al. [96] modelled NFVI-PoPs that can host multiple VMs, which in turn can host multiple VNFs. This realistic mapping is coupled with a distributed algorithm where each VM makes decisions about the creation or destruction of VNFs based on information about itself and immediate neighbours. Their distributed technique prioritises small requests with high prices. It also tries to allocate incoming requests into existing VNFs before installing new ones, thus maximising the number of VNFs on a single VM. The authors prove that the distributed algorithm achieves asymptotically near-optimal performance regarding *cost reduction*, which is its main objective. Finally, Xie et al. [97] designed an optimal placement algorithm aiming to *increase revenue* for embedding VNFs in a given path. To jointly solve the VNF-FGE problem, it is then invoked for incoming requests, using the shortest path between the ingress and egress nodes. The optimal placement of VNFs is calculated using an augmented graph, which is formulated so that the shortest path in this graph results in the optimal allocation of VNFs.

Having made an orthogonal analysis of the area of VNF-FG embedding and described the state-of-the-art, we now discuss identified challenges and shortcomings.

6. Challenges and shortcomings

The development of (sub-)optimal solutions to the VNF-FGE problem has paramount importance for the successful NFV adoption. Based on our review, we identified challenges and shortcomings that must be overcome by future work, detailed next with recommendations for future research efforts.

Integration with VNF-CC and VNF-SCH. One of the challenges to be overcome is how to provide VNF-FGE algorithms that also work together with algorithms for the VNF-CC and VNF-SCH problems. Few research efforts [10,13,72,79] introduce approaches that can tackle either VNF-CC or VNF-SCH along with VNF-FGE. However, no reviewed work tackle these three problems altogether.

ETSI-compliant interfaces. Most approaches introduce a mathematical formalisation of the VNF-FGE problem, propose a solution and provide an empirical evaluation of its performance. However, the compliance of these solutions with ETSI's standards remains insufficiently investigated. The NFV architecture introduced by ETSI [19] contains interfaces with well defined operations [2], which are typically ignored by current research. Furthermore, despite the fact that the VNF-FGE problem is composed of two sub-problems, distinct algorithms that tackle the placement and chaining problems cannot be easily integrated to form a VNF-FGE solution. Fully embracing ETSI's standards might ease such efforts, allowing algorithms that attack the VNF-FGE problem to be exchangeable in real-life applications.

Lifecycle management of VNFs. Future solutions to the VNF-FGE problem should fully manage the lifecycle of VNFs. Currently, most research efforts are concerned with modelling the instantiation of VNF instances. However, other operations such as software upgrade, instance modification [125], instance migration [41], instance scaling up/down [99] and instance termination are commonly neglected. The many challenges related to the lifecycle management of VNFs have been discussed at length in [126,127].

Lack of factors and their relationships. The absence of network and host characteristics such as latency, memory and storage capacity complicates the adoption of approaches by network operators as they do not reflect the network accurately. Moreover, not correctly modelling the relationship between factors leads to unrealistic models that do not entirely capture the behaviour of VNFs and the network. Except for one reviewed work [92], current research efforts assume that factors are entirely independent from each other, despite empirical observations providing evidence otherwise [128,129].

Lack of user preferences. Allowing users to inform their preferences over how much use of specific physical machines, network links and VNFs should be modelled into VNF-FGE approaches. We advocate that allowing the network operator to express preferences over parameters should improve the adoption of a VNF-FGE solution, because the network operator could tune the algorithm to her specific needs without diving deeply on its internals. Currently, a few research efforts take as input the user preferences over some characteristics but, in general, do not provide complete control over the algorithm [13,41,54,58,72,99].

Comparison of proposed algorithms. Most reviewed works provide an empirical evaluation of the proposed solution, which is commonly compared with variants of the introduced algorithm and with the optimal solution of the ILP or MIP formulation. However, despite tackling the same problem, the comparison with previous work is uncommon [22,61,67]. Given that algorithms aim to optimise the allocation of specific factors (*e.g.* minimise bandwidth usage), the empirical evaluation should provide evidence of the non-optimised factors (*e.g.* CPU and memory usage), thus allowing researchers to evaluate the behaviour of the proposed algorithm entirely. Moreover, there is no common set of experiments in the literature to evaluate the performance of VNF-FGE algorithms, *i.e.* an evaluation methodology. Therefore, it is mostly impossible to compare and understand the trade-off between algorithms unless the authors have provided an explicit comparison. Finally, we advocate that to compare VNF-FGE algorithms, future research efforts should focus on establishing common ground by: (i) using the same set of facets and their non-linear relationship [129]; and (ii) creating an evaluation methodology, specifying test scenarios, scalability patterns and mandatory measurements.

7. Conclusion

In this paper, we surveyed the existing literature that proposed approaches in the context of the VNF-FGE problem. Existing work was classified according to a proposed taxonomy, composed of five facets, namely behaviour, allocation task, objective, factors and technique style. These facets not only allow to classify research proposals, but also understand their key characteristics, thus facilitating their comparison. The facets also serve for authors of future approaches as a checklist of aspects to be considered. Finally, we outlined shortcomings of current research efforts as well as open challenges, leading to future directions in the area.

CRediT authorship contribution statement

Frederico Schardong: Software, Investigation, Data curation, Writing - original draft, Visualization. **Ingrid Nunes:** Methodology, Validation, Writing - review & editing, Visualization, Supervision. **Alberto Schaeffer-Filho:** Conceptualization, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The authors of this work would like to thank the financial support of CNPq, Brazil for research grants ref. 313357/2018-8, 428157/2018-1, 311088/2015-5 and 407899/2016-2.

References

[1] R. Guerzoni, et al., Network functions virtualisation: an introduction, benefits, enablers, challenges and call for action, introductory white paper, in: SDN and OpenFlow World Congress, 2012, pp. 5–7, available at https://portal.etsi.org/nfv/nfv_white_paper.pdf.

[2] ETSI ISG NFV, Network functions virtualisation (NFV): Architectural framework, 2013, URL http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_NFV002v010201p.pdf.

[3] H. Moens, F. De Turck, VNF-P: A model for efficient placement of virtualized network functions, in: CNSM, IEEE, 2014, pp. 418–423, http://dx.doi.org/10.1109/CNSM.2014.7014205.

[4] W. John, K. Pentikousis, G. Agapiou, E. Jacob, M. Kind, A. Manzalini, F. Risso, D. Staessens, R. Steinert, C. Meirosu, Research directions in network service chaining, in: SDN4FNS, IEEE, 2013, pp. 1–7, http://dx.doi.org/10.1109/SDN4FNS.2013.6702549.

[5] J.G. Herrera, J.F. Botero, Resource allocation in NFV: A comprehensive survey, IEEE Trans. Netw. Serv. Manag. 13 (3) (2016) 518–532, http://dx.doi.org/10.1109/TNSM.2016.2598420.

[6] R. Cohen, L. Lewin-Eytan, J.S. Naor, D. Raz, Near optimal placement of virtual network functions, in: INFOCOM, IEEE, 2015, pp. 1346–1354, http://dx.doi.org/10.1109/INFOCOM.2015.7218511.

[7] H.D. Chantre, N.L. da Fonseca, Redundant placement of virtualized network functions for LTE evolved multimedia broadcast multicast services, in: ICC, IEEE, 2017, pp. 1–7, http://dx.doi.org/10.1109/ICC.2017.7996870.

[8] S. Jiao, X. Zhang, S. Yu, X. Song, Z. Xu, Joint virtual network function selection and traffic steering in telecom networks, in: GLOBECOM, 2017, pp. 1–7, http://dx.doi.org/10.1109/GLOCOM.2017.8254652.

[9] M.-T. Thai, Y.-D. Lin, Y.-C. Lai, A joint network and server load balancing algorithm for chaining virtualized network functions, in: ICC, IEEE, 2016, pp. 1–6, http://dx.doi.org/10.1109/ICC.2016.7510712.

[10] H.A. Alameddine, L. Qu, C. Assi, Scheduling service function chains for ultra-low latency network services, in: CNSM, IEEE, 2017, pp. 1–9, http://dx.doi.org/10.23919/CNSM.2017.8256017.

[11] M.C. Luizelli, L.R. Bays, L.S. Buriol, M.P. Barcellos, L.P. Gaspary, Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions, in: IM, IEEE, 2015, pp. 98–106, http://dx.doi.org/10.1109/INM.2015.7140281.

[12] O. Soualah, M. Mechtri, C. Ghribi, D. Zeghlache, A link failure recovery algorithm for virtual network function chaining, in: IM, IEEE, 2017, pp. 213–221, http://dx.doi.org/10.23919/INM.2017.7987282.

[13] F. Schardong, I. Nunes, A. Schaeffer-Filho, A distributed NFV orchestrator based on BDI reasoning, in: IM, IEEE, 2017, pp. 107–115, http://dx.doi.org/10.23919/INM.2017.7987270.

[14] N.F.S. de Sousa, D.A.L. Perez, R.V. Rosa, M.A. Santos, C.E. Rothenberg, Network service orchestration: A survey, Comput. Commun. (2019) http://dx.doi.org/10.1016/j.comcom.2019.04.008.

[15] Y. Li, M. Chen, Software-defined network function virtualization: A survey, IEEE Access 3 (2015) 2542–2553, http://dx.doi.org/10.1109/ACCESS.2015.2499271.

[16] X. Li, C. Qian, A survey of network function placement, in: CCNC, IEEE, 2016, pp. 948–953, http://dx.doi.org/10.1109/CCNC.2016.7444915.

[17] D. Bhamare, R. Jain, M. Samaka, A. Erbad, A survey on service function chaining, J. Netw. Comput. Appl. 75 (2016) 138–155, http://dx.doi.org/10.1016/j.jnca.2016.09.001.

[18] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, R. Boutaba, Network function virtualization: State-of-the-art and research challenges, IEEE Commun. Surv. Tutor. 18 (1) (2015) 236–262, http://dx.doi.org/10.1109/COMST.2015.2477041.

[19] ETSI ISG NFV, ETSI GS NFV 003 V1.2.1: Network functions virtualisation (NFV); terminology for main concepts in NFV, 2014, URL http://www.etsi.org/deliver/etsi_gs/NFV/001_099/003/01.02.01_60/gs_NFV003v010201p.pdf.

[20] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J.J. Ramos-Munoz, J. Lorca, J. Folgueira, Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges, IEEE Commun. Mag. 55 (5) (2017) 80–87, http://dx.doi.org/10.1109/MCOM.2017.1600935.

[21] E.J. Scheid, C.C. Machado, M.F. Franco, R.L. dos Santos, R.P. Pfitscher, A.E. Schaeffer-Filho, L.Z. Granville, INSpIRE: Integrated NFV-based intent refinement environment, in: IM, IEEE, 2017, pp. 186–194, http://dx.doi.org/10.23919/INM.2017.7987279.

[22] M.T. Beck, J.F. Botero, Coordinated allocation of service function chains, in: GLOBECOM, IEEE, 2015, pp. 1–6, http://dx.doi.org/10.1109/GLOCOM.2015.7417401.

[23] A. Haider, R. Potter, A. Nakao, Challenges in resource allocation in network virtualization, in: 20th ITC Specialist Seminar, vol. 18, no. 2009, ITC, 2009.

[24] A. Fischer, J.F. Botero, M.T. Beck, H. De Meer, X. Hesselbach, Virtual network embedding: A survey, IEEE Commun. Surv. Tutor. 15 (4) (2013) 1888–1906, http://dx.doi.org/10.1109/SURV.2013.013013.00155.

[25] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, S. Davy, Design and evaluation of algorithms for mapping and scheduling of virtual network functions, in: NETSOFT, IEEE, 2015, pp. 1–9, http://dx.doi.org/10.1109/NETSOFT.2015.7116120.

[26] B. Kitchenham, Procedures for Performing Systematic Reviews, Keele University, UK, 2004.

[27] T.-W. Kuo, B.-H. Liou, K.C.-J. Lin, M.-J. Tsai, Deploying chains of virtual network functions: On the relation between link and server usage, in: INFOCOM, IEEE, 2016, pp. 1–9, http://dx.doi.org/10.1109/INFOCOM.2016.7524565.

[28] Y. Li, L.T.X. Phan, B.T. Loo, Network functions virtualization with soft real-time guarantees, in: INFOCOM, IEEE, 2016, pp. 1–9, http://dx.doi.org/10.1109/INFOCOM.2016.7524563.

[29] H. Feng, J. Llorca, A.M. Tulino, D. Raz, A.F. Molisch, Approximation algorithms for the NFV service distribution problem, in: INFOCOM, IEEE, 2017, pp. 1–9, http://dx.doi.org/10.1109/INFOCOM.2017.8057039.

[30] X. Lin, D. Guo, Y. Shen, G. Tang, B. Ren, Dag-sfc: Minimize the embedding cost of sfc with parallel vnfs, in: Proceedings of the 47th International Conference on Parallel Processing, 2018, pp. 1–10, http://dx.doi.org/10.1145/3225058.3225111.

[31] B. Ren, D. Guo, G. Tang, X. Lin, Y. Qin, Optimal service function tree embedding for NFV enabled multicast, in: 2018 IEEE 38th International Conference on Distributed Computing Systems, ICDCS, IEEE, 2018, pp. 132–142, http://dx.doi.org/10.1109/ICDCS.2018.00023.

[32] B. Ren, D. Guo, Y. Shen, G. Tang, X. Lin, Embedding service function tree with minimum cost for NFV-enabled multicast, IEEE J. Sel. Areas Commun. 37 (5) (2019) 1085–1097, http://dx.doi.org/10.1109/JSAC.2019.2906764.

[33] M. Xia, M. Shirazipour, Y. Zhang, H. Green, A. Takacs, Network function placement for NFV chaining in packet/optical datacenters, J. Lightwave Technol. 33 (8) (2015) 1565–1570, http://dx.doi.org/10.1109/JLT.2015.2388585.

[34] Z. Xu, W. Liang, A. Galis, Y. Ma, Q. Xia, W. Xu, Throughput optimization for admitting NFV-enabled requests in cloud networks, Comput. Netw. 143 (2018) 15–29, http://dx.doi.org/10.1016/j.comnet.2018.06.015.

[35] M.T. Beck, J.F. Botero, Scalable and coordinated allocation of service function chains, Comput. Commun. 102 (2017) 78–88, http://dx.doi.org/10.1016/j.comcom.2016.09.010.

[36] Z. Ye, X. Cao, J. Wang, H. Yu, C. Qiao, Joint topology design and mapping of service function chains for efficient, scalable, and reliable network functions virtualization, IEEE Netw. 30 (3) (2016) 81–87, http://dx.doi.org/10.1109/MNET.2016.7474348.

[37] N.A. El-Sherbeny, Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods, J. King Saud Univ.-Sci. 22 (3) (2010) 123–131, http://dx.doi.org/10.1016/j.jksus.2010.03.002.

[38] G. Zobolas, C.D. Tarantilis, G. Ioannou, Exact, heuristic and meta-heuristic algorithms for solving shop scheduling problems, in: Metaheuristics for Scheduling in Industrial and Manufacturing Applications, Springer, 2008, pp. 1–40, http://dx.doi.org/10.1007/978-3-540-78985-7_1.

[39] F. Glover, L.H. Cox, J.P. Kelly, R. Patil, Exact, heuristic and metaheuristic methods for confidentiality protection by controlled tabular adjustment, Int. J. Oper. Res. 5 (2) (2008) 117–128.

[40] X.-S. Yang, Nature-Inspired Metaheuristic Algorithms, Luniver press, ISBN: 978-1-905986-10-1, 2010.

[41] P.-W. Chi, Y.-C. Huang, C.-L. Lei, Efficient NFV deployment in data center networks, in: ICC, IEEE, 2015, pp. 5290–5295, http://dx.doi.org/10.1109/ICC.2015.7249164.

[42] Y. Sang, B. Ji, G.R. Gupta, X. Du, L. Ye, Provably efficient algorithms for joint placement and allocation of virtual network functions, in: INFOCOM, IEEE, 2017, pp. 1–9, http://dx.doi.org/10.1109/INFOCOM.2017.8057036.

[43] A. Tomassilli, F. Giroire, N. Huin, S. Pérennes, Provably efficient algorithms for placement of service function chains with ordering constraints, in: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, IEEE, 2018, pp. 774–782, http://dx.doi.org/10.1109/INFOCOM.2018.8486275.

[44] G. Sallam, B. Ji, Joint placement and allocation of virtual network functions with budget and capacity constraints, in: IEEE INFOCOM 2019-IEEE Conference on Computer Communications, IEEE, 2019, pp. 523–531, http://dx.doi.org/10.1109/INFOCOM.2019.8737400.

[45] L. Yala, P.A. Frangoudis, A. Ksentini, Latency and availability driven VNF placement in a MEC-NFV environment, in: 2018 IEEE Global Communications Conference, GLOBECOM, IEEE, 2018, pp. 1–7, http://dx.doi.org/10.1109/GLOCOM.2018.8647858.

[46] D.M. Manias, M. Jammal, H. Hawilo, A. Shami, P. Heidari, A. Larabi, R. Brunner, Machine learning for performance-aware virtual network function placement, in: 2019 IEEE Global Communications Conference, GLOBECOM, IEEE, 2019, pp. 1–6, http://dx.doi.org/10.1109/GLOBECOM38437.2019.9013246.

[47] Z. Xu, W. Liang, A. Galis, Y. Ma, Throughput maximization and resource optimization in NFV-enabled networks, in: ICC, IEEE, 2017, pp. 1–7, http://dx.doi.org/10.1109/ICC.2017.7996514.

[48] N. Huin, B. Jaumard, F. Giroire, Optimization of network service chain provisioning, in: ICC, IEEE, 2017, pp. 1–7, http://dx.doi.org/10.1109/ICC.2017.7997198.

[49] A. Tomassilli, N. Huin, F. Giroire, B. Jaumard, Resource requirements for reliable service function chaining, in: 2018 IEEE International Conference on Communications, ICC, IEEE, 2018, pp. 1–7, http://dx.doi.org/10.1109/ICC.2018.8422774.

[50] Z. Ye, X. Cao, C. Qiao, Joint topology design and mapping of service function chains in network function virtualization, in: GLOBECOM, IEEE, 2016, pp. 1–6, http://dx.doi.org/10.1109/GLOCOM.2016.7841939.

[51] A. Gupta, B. Jaumard, M. Tornatore, B. Mukherjee, Service chain (SC) mapping with multiple SC instances in a wide area network, in: GLOBECOM, 2017, pp. 1–6, http://dx.doi.org/10.1109/GLOCOM.2017.8254731.

[52] J. Kong, I. Kim, X. Wang, Q. Zhang, H.C. Cankaya, W. Xie, T. Ikeuchi, J.P. Jue, Guaranteed-availability network function virtualization with network protection and VNF replication, in: GLOBECOM, 2017, pp. 1–6, http://dx.doi.org/10.1109/GLOCOM.2017.8254730.

[53] Z. Chen, S. Zhang, C. Wang, Z. Qian, M. Xiao, J. Wu, I. Jawhar, A novel algorithm for NFV chain placement in edge computing environments, in: 2018 IEEE Global Communications Conference, GLOBECOM, IEEE, 2018, pp. 1–6, http://dx.doi.org/10.1109/GLOCOM.2018.8647371.

[54] X. Song, X. Zhang, S. Yu, S. Jiao, Z. Xu, Resource-efficient virtual network function placement in operator networks, in: GLOBECOM, 2017, pp. 1–7, http://dx.doi.org/10.1109/GLOCOM.2017.8254492.

[55] J.-J. Kuo, S.-H. Shen, H.-Y. Kang, D.-N. Yang, M.-J. Tsai, W.-T. Chen, Service chain embedding with maximum flow in software defined network and application to the next-generation cellular network architecture, in: INFOCOM, IEEE, 2017, pp. 1–9, http://dx.doi.org/10.1109/INFOCOM.2017.8057229.

[56] Z. Xu, X. Zhang, S. Yu, J. Zhang, Energy-efficient virtual network function placement in telecom networks, in: 2018 IEEE International Conference on Communications, ICC, IEEE, 2018, pp. 1–7, http://dx.doi.org/10.1109/ICC.2018.8422879.

[57] X. Zhong, Y. Wang, X. Qiu, Cost-aware service function chaining with reliability guarantees in NFV-enabled inter-DC network, in: 2019 IFIP/IEEE Symposium on Integrated Network and Service Management, IM, IEEE, 2019, pp. 304–311.

[58] M. Wang, B. Cheng, W. Feng, J. Chen, An efficient service function chain placement algorithm in a MEC-NFV environment, in: 2019 IEEE Global Communications Conference, GLOBECOM, IEEE, 2019, pp. 1–6, http://dx.doi.org/10.1109/GLOBECOM38437.2019.9013235.

[59] H. Chen, S. Xu, X. Wang, Y. Zhao, K. Li, Y. Wang, W. Wang, et al., Towards optimal outsourcing of service function chain across multiple clouds, in: ICC, IEEE, 2016, pp. 1–7, http://dx.doi.org/10.1109/ICC.2016.7510996.

[60] P. Vizarreta, M. Condoluci, C.M. Machuca, T. Mahmoodi, W. Kellerer, QoS-driven function placement reducing expenditures in NFV deployments, in: ICC, IEEE, 2017, pp. 1–7, http://dx.doi.org/10.1109/ICC.2017.7996513.

[61] N. Tastevin, M. Obadia, M. Bouet, A graph approach to placement of service functions chains, in: IM, IEEE, 2017, pp. 134–141, http://dx.doi.org/10.23919/INM.2017.7987273.

[62] F. Carpio, S. Dhahri, A. Jukan, VNF Placement with replication for load balancing in NFV networks, in: ICC, IEEE, 2017, pp. 1–6, http://dx.doi.org/10.1109/ICC.2017.7996515.

[63] F.B. Jemaa, G. Pujolle, M. Pariente, QoS-aware VNF placement optimization in edge-central carrier cloud architecture, in: GLOBECOM, IEEE, 2016, pp. 1–7, http://dx.doi.org/10.1109/GLOCOM.2016.7842188.

[64] W. Zhou, Y. Yang, M. Xu, H. Chen, Accommodating dynamic traffic immediately: A VNF placement approach, in: ICC 2019-2019 IEEE International Conference on Communications, ICC, IEEE, 2019, pp. 1–6, http://dx.doi.org/10.1109/ICC.2019.8761554.

[65] C. You, et al., Efficient load balancing for the VNF deployment with placement constraints, in: ICC 2019-2019 IEEE International Conference on Communications, ICC, IEEE, 2019, pp. 1–6, http://dx.doi.org/10.1109/ICC.2019.8761564.

[66] J. Li, W. Shi, P. Yang, X. Shen, On dynamic mapping and scheduling of service function chains in SDN/NFV-enabled networks, in: 2019 IEEE Global Communications Conference, GLOBECOM, IEEE, 2019, pp. 1–6, http://dx.doi.org/10.1109/GLOBECOM38437.2019.9013429.

[67] X. Zhang, C. Wu, Z. Li, F.C. Lau, Proactive VNF provisioning with multi-timescale cloud resources: Fusing online learning and online optimization, in: INFOCOM, IEEE, 2017, pp. 1–9, http://dx.doi.org/10.1109/INFOCOM.2017.8057118.

[68] J. Sun, F. Liu, M. Ahmed, Y. Li, Efficient virtual network function placement for Poisson arrived traffic, in: ICC 2019-2019 IEEE International Conference on Communications, ICC, IEEE, 2019, pp. 1–7, http://dx.doi.org/10.1109/ICC.2019.8761609.

[69] M. Jia, W. Liang, M. Huang, Z. Xu, Y. Ma, Throughput maximization of NFV-enabled unicasting in software-defined networks, in: GLOBECOM, IEEE, 2017, pp. 1–6, http://dx.doi.org/10.1109/GLOCOM.2017.8254756.

[70] G. Sallam, G.R. Gupta, B. Li, B. Ji, Shortest path and maximum flow problems under service function chaining constraints, in: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, IEEE, 2018, pp. 2132–2140, http://dx.doi.org/10.1109/INFOCOM.2018.8485996.

[71] L. Gao, G.N. Rouskas, On congestion minimization for service chain routing problems, in: ICC 2019-2019 IEEE International Conference on Communications, ICC, IEEE, 2019, pp. 1–6, http://dx.doi.org/10.1109/ICC.2019.8761660.

[72] X. Huang, S. Bian, X. Gao, W. Wu, Z. Shao, Y. Yang, Online VNF chaining and scheduling with prediction: Optimality and trade-offs, in: 2019 IEEE Global Communications Conference, GLOBECOM, IEEE, 2019, pp. 1–6, http://dx.doi.org/10.1109/GLOBECOM38437.2019.9013961.

[73] J. Zhou, P. Hong, J. Pei, Multi-task deep learning based dynamic service function chains routing in SDN/NFV-enabled networks, in: ICC 2019-2019 IEEE International Conference on Communications, ICC, IEEE, 2019, pp. 1–6, http://dx.doi.org/10.1109/ICC.2019.8761116.

[74] W. Ma, C. Medina, D. Pan, Traffic-aware placement of NFV middleboxes, in: GLOBECOM, IEEE, 2015, pp. 1–6, http://dx.doi.org/10.1109/GLOCOM.2015.7417851.

[75] B. Feng, G. Li, G. Li, H. Zhou, H. Zhang, S. Yu, Efficient mappings of service function chains at terrestrial-satellite hybrid cloud networks, in: 2018 IEEE Global Communications Conference, GLOBECOM, IEEE, 2018, pp. 1–6, http://dx.doi.org/10.1109/GLOCOM.2018.8647330.

[76] A. Mohamad, H.S. Hassanein, On demonstrating the gain of SFC placement with VNF sharing at the edge, in: 2019 IEEE Global Communications Conference, GLOBECOM, IEEE, 2019, pp. 1–6, http://dx.doi.org/10.1109/GLOBECOM38437.2019.9014106.

[77] W. Ma, O. Sandoval, J. Beltran, D. Pan, N. Pissinou, Traffic aware placement of interdependent NFV middleboxes, in: INFOCOM, IEEE, 2017, pp. 1–9, http://dx.doi.org/10.1109/INFOCOM.2017.8056993.

[78] M. Liu, G. Feng, J. Zhou, S. Qin, Joint two-tier network function parallelization on multicore platform, in: 2018 IEEE Global Communications Conference, GLOBECOM, 2018, pp. 1–7, http://dx.doi.org/10.1109/GLOCOM.2018.8647513.

[79] M. Jalalitabar, E. Guler, G. Luo, L. Tian, X. Cao, Dependence-aware service function chain design and mapping, in: GLOBECOM, 2017, pp. 1–6, http://dx.doi.org/10.1109/GLOCOM.2017.8254485.

[80] M. Huang, W. Liang, Y. Ma, S. Guo, Throughput maximization of delay-sensitive request admissions via virtualized network function placements and migrations, in: 2018 IEEE International Conference on Communications, ICC, IEEE, 2018, pp. 1–7, http://dx.doi.org/10.1109/ICC.2018.8422337.

[81] T. Wen, H. Yu, G. Sun, L. Liu, Network function consolidation in service function chaining orchestration, in: ICC, IEEE, 2016, pp. 1–6, http://dx.doi.org/10.1109/ICC.2016.7510679.

[82] F. Aklamanu, S. Randriamasy, E. Renault, Utility and a*-based algorithm for network slice placement and chaining, in: 2019 IEEE Global Communications Conference, GLOBECOM, IEEE, 2019, pp. 1–6, http://dx.doi.org/10.1109/GLOBECOM38437.2019.9013820.

[83] S.Q. Zhang, A. Tizghadam, B. Park, H. Bannazadeh, A. Leon-Garcia, Joint NFV placement and routing for multicast service on sdn, in: NOMS, IEEE, 2016, pp. 333–341, http://dx.doi.org/10.1109/NOMS.2016.7502829.

[84] H. Feng, J. Llorca, A.M. Tulino, A.F. Molisch, Optimal dynamic cloud network control, in: 2016 IEEE International Conference on Communications, ICC, IEEE, 2016, pp. 1–7, http://dx.doi.org/10.1109/ICC.2016.7511591.

[85] K. Xie, X. Zhou, T. Semong, S. He, Multi-source multicast routing with QoS constraints in network function virtualization, in: ICC 2019-2019 IEEE International Conference on Communications, ICC, IEEE, 2019, pp. 1–6, http://dx.doi.org/10.1109/ICC.2019.8761957.

[86] Y. Liu, J. Pei, P. Hong, D. Li, Cost-efficient virtual network function placement and traffic steering, in: ICC 2019-2019 IEEE International Conference on Communications, ICC, IEEE, 2019, pp. 1–6, http://dx.doi.org/10.1109/ICC.2019.8762060.

[87] B. Spinnewyn, J.F. Botero, C. Donato, S. Latré, Effective NFV orchestration for wide-ranging services across heterogeneous cloud networks, in: 2019 IFIP/IEEE Symposium on Integrated Network and Service Management, IM, IEEE, 2019, pp. 107–115.

[88] J. Fan, C. Guan, Y. Zhao, C. Qiao, Availability-aware mapping of service function chains, in: INFOCOM, IEEE, 2017, pp. 1–9, http://dx.doi.org/10.1109/INFOCOM.2017.8057153.

[89] A. Varasteh, M. De Andrade, C.M. Machuca, L. Wosinska, W. Kellerer, Power-aware virtual network function placement and routing using an abstraction technique, in: 2018 IEEE Global Communications Conference, GLOBECOM, IEEE, 2018, pp. 1–7, http://dx.doi.org/10.1109/GLOCOM.2018.8647538.

[90] X. Fei, F. Liu, H. Xu, H. Jin, Adaptive VNF scaling and flow routing with proactive demand prediction, in: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, IEEE, 2018, pp. 486–494, http://dx.doi.org/10.1109/INFOCOM.2018.8486320.

[91] B. Anwer, T. Benson, N. Feamster, D. Levin, Programming slick network functions, in: Proceedings of the 1st Acm Sigcomm Symposium on Software Defined Networking Research, 2015, pp. 1–13, http://dx.doi.org/10.1145/2774993.2774998.

[92] W. Racheg, N. Ghrada, M.F. Zhani, Profit-driven resource provisioning in NFV-based environments, in: ICC, IEEE, 2017, pp. 1–7, http://dx.doi.org/10.1109/ICC.2017.7997163.

[93] Q. Sun, P. Lu, W. Lu, Z. Zhu, Forecast-assisted NFV service chain deployment based on affiliation-aware VNF placement, in: GLOBECOM, IEEE, 2016, pp. 1–6, http://dx.doi.org/10.1109/GLOCOM.2016.7841846.

[94] Y. Ma, W. Liang, M. Huang, S. Guo, Profit maximization of NFV-enabled request admissions in SDNs, in: 2018 IEEE Global Communications Conference, GLOBECOM, IEEE, 2018, pp. 1–7, http://dx.doi.org/10.1109/GLOCOM.2018.8647455.

[95] Y. Ma, W. Liang, Z. Xu, Online revenue maximization in NFV-enabled SDNs, in: 2018 IEEE International Conference on Communications, ICC, IEEE, 2018, pp. 1–7, http://dx.doi.org/10.1109/ICC.2018.8422333.

[96] X. Chen, W. Ni, I.B. Collings, X. Wang, S. Xu, Distributed placement and online optimization of virtual machines for network service chains, in: 2018 IEEE International Conference on Communications, ICC, IEEE, 2018, pp. 1–6, http://dx.doi.org/10.1109/ICC.2018.8422336.

[97] Y. Xie, S. Wang, Y. Dai, Provable algorithm for virtualised network function chain placement in dynamic environment, in: 2019 IEEE Global Communications Conference, GLOBECOM, IEEE, 2019, pp. 1–6, http://dx.doi.org/10.1109/GLOBECOM38437.2019.9013582.

[98] H.R. Khezri, P.A. Moghadam, M.K. Farshbafan, V. Shah-Mansouri, H. Kebriaei, D. Niyato, Deep reinforcement learning for dynamic reliability aware NFV-based service provisioning, in: 2019 IEEE Global Communications Conference, GLOBECOM, IEEE, 2019, pp. 1–6, http://dx.doi.org/10.1109/GLOBECOM38437.2019.9013214.

[99] W. Rankothge, J. Ma, F. Le, A. Russo, J. Lobo, Towards making network function virtualization a cloud computing service, in: IM, IEEE, 2015, pp. 89–97, http://dx.doi.org/10.1109/INM.2015.7140280.

[100] M. Rost, S. Schmid, On the hardness and inapproximability of virtual network embeddings, IEEE/ACM Trans. Netw. 28 (2) (2020) 791–803, http://dx.doi.org/10.1109/TNET.2020.2975646.

[101] J. Lofberg, YALMIP: A toolbox for modeling and optimization in MATLAB, in: CACSD, IEEE, 2004, pp. 284–289, http://dx.doi.org/10.1109/CACSD.2004.1393890.

[102] N. Karmarkar, R.M. Karp, An efficient approximation scheme for the one-dimensional bin-packing problem, in: SFCS, IEEE, 1982, pp. 312–320, http://dx.doi.org/10.1109/SFCS.1982.61.

[103] S. Khuller, A. Moss, J.S. Naor, The budgeted maximum coverage problem, Inf. Process. Lett. 70 (1) (1999) 39–45, http://dx.doi.org/10.1016/S0020-0190(99)00031-9.

[104] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95 - International Conference on Neural Networks, vol. 4, 1995, pp. 1942–1948, http://dx.doi.org/10.1109/ICNN.1995.488968.

[105] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (Oct) (2011) 2825–2830.

[106] H. Hawilo, M. Jammal, A. Shami, Network function virtualization-aware orchestrator for service function chaining placement in the cloud, IEEE J. Sel. Areas Commun. 37 (3) (2019) 643–655, http://dx.doi.org/10.1109/JSAC.2019.2895226.

[107] Y. Wang, X. Zhang, L. Fan, S. Yu, R. Lin, Segment routing optimization for VNF chaining, in: ICC 2019-2019 IEEE International Conference on Communications, ICC, IEEE, 2019, pp. 1–7, http://dx.doi.org/10.1109/ICC.2019.8761103.

[108] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W. Savelsbergh, P.H. Vance, Branch-and-price: Column generation for solving huge integer programs, Oper. Res. 46 (3) (1998) 316–329, http://dx.doi.org/10.1287/opre.46.3.316.

[109] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, vol. 16, John Wiley & Sons, ISBN: 978-0-471-87339-6, 2001.

[110] L. Rabiner, B. Juang, An introduction to hidden Markov models, IEEE ASSP Mag. 3 (1) (1986) 4–16, http://dx.doi.org/10.1109/MASSP.1986.1165342.

[111] G.D. Forney, The viterbi algorithm, Proc. IEEE 61 (3) (1973) 268–278, http://dx.doi.org/10.1109/PROC.1973.9030.

[112] J.A. Nelder, R. Mead, A simplex method for function minimization, Comput. J. 7 (4) (1965) 308–313, http://dx.doi.org/10.1093/comjnl/7.4.308.

[113] U. Brandes, On variants of shortest-path betweenness centrality and their generic computation, Social Networks 30 (2) (2008) 136–145, http://dx.doi.org/10.1016/j.socnet.2007.11.001.

[114] M. Zinkevich, Online convex programming and generalized infinitesimal gradient ascent, in: ICML, 2003, pp. 928–936, http://dx.doi.org/10.5555/3041838.3041955.

[115] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680, http://dx.doi.org/10.1126/science.220.4598.671.

[116] P. Festa, Complexity analysis and optimization of the shortest path tour problem, Optim. Lett. 6 (1) (2012) 163–175, http://dx.doi.org/10.1007/s11590-010-0258-y.

[117] F. Glover, Tabu search—part I, ORSA J. Comput. 1 (3) (1989) 190–206, http://dx.doi.org/10.1287/ijoc.1.3.190.

[118] A.S. Rao, M.P. Georgeff, et al., BDI Agents: From theory to practice, in: ICMAS, vol. 95, 1995, pp. 312–319.

[119] P.E. Hart, N.J. Nilsson, B. Raphael, A formal basis for the heuristic determination of minimum cost paths, IEEE Trans. Syst. Sci. Cybern. 4 (2) (1968) 100–107, http://dx.doi.org/10.1109/TSSC.1968.300136.

[120] S.Q. Zhang, Q. Zhang, H. Bannazadeh, A. Leon-Garcia, Network function virtualization enabled multicast routing on SDN, in: ICC, IEEE, 2015, pp. 5595–5601, http://dx.doi.org/10.1109/ICC.2015.7249214.

[121] C.B. Browne, E. Powley, D. Whitehouse, S.M. Lucas, P.I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, S. Colton, A survey of Monte Carlo tree search methods, IEEE Trans. Comput. Intell. AI Games 4 (1) (2012) 1–43, http://dx.doi.org/10.1109/TCIAIG.2012.2186810.

[122] C.J.C.H. Watkins, Learning from Delayed Rewards (Ph.D. thesis), Cambridge University, King's College, 1989.

[123] H.B. McMahan, A unified view of regularized dual averaging and mirror descent with implicit updates, 2010, arXiv preprint arXiv:1009.3240.

[124] A. Fumi, A. Pepe, L. Scarabotti, M.M. Schiraldi, Fourier analysis for demand forecasting in a fashion company, Int. J. Eng. Bus. Manage. 5 (2013) 30, http://dx.doi.org/10.5772/56839.

[125] S. Dobson, D. Hutchison, A. Mauthe, A. Schaeffer-Filho, P. Smith, J.P.G. Sterbenz, Self-organization and resilience for networked systems: Design principles and open research issues, Proc. IEEE 107 (4) (2019) 819–834.

[126] G. Venâncio, V.F. Garcia, L. da Cruz Marcuzzo, T.N. Tavares, M.F. Franco, L. Bondan, A.E. Schaeffer-Filho, C.R. Paula dos Santos, L.Z. Granville, E.P. Duarte Jr., Beyond VNFM: Filling the gaps of the ETSI VNF manager to fully support VNF life cycle operations, Int. J. Netw. Manage., n/a e2068, http://dx.doi.org/10.1002/nem.2068.

[127] L. Bondan, M.F. Franco, L. Marcuzzo, G. Venancio, R.L. Santos, R.J. Pfitscher, E.J. Scheid, B. Stiller, F. De Turck, E.P. Duarte, A.E. Schaeffer-Filho, C.R.P. d. Santos, L.Z. Granville, FENDE: Marketplace-based distribution, execution, and life cycle management of VNFs, IEEE Commun. Mag. 57 (1) (2019) 13–19.

[128] R. Birke, L.Y. Chen, E. Smirni, Multi-resource characterization and their (in)dependencies in production datacenters, in: NOMS, IEEE, 2014, pp. 1–6, http://dx.doi.org/10.1109/NOMS.2014.6838300.

[129] M.C. Luizelli, D. Raz, Y. Sa'ar, J. Yallouz, The actual cost of software switching for NFV chaining, in: IM, IEEE, 2017, pp. 335–343, http://dx.doi.org/10.23919/INM.2017.7987296.

Frederico Schardong is a Lecturer at the Instituto Federal do Rio Grande do Sul (IFRS), Brazil, currently pursuing his Ph.D. in Computer Science at the Universidade Federal de Santa Catarina (UFSC), in Brazil. He achieved his M.Sc. Degree in Computer Science in Universidade Federal do Rio Grande do Sul (UFRGS), in 2018. His research interests include network security and resilience, Network Function Virtualization (NFV), machine learning and multi-agent systems.

Ingrid Nunes is an Associate Professor at the Informatics Institute, Universidade Federal do Rio Grande do Sul (UFRGS), Brazil, and holds a research productivity (PQ) fellowship Level 2 granted by CNPq. She completed her Ph.D. in Informatics at the Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Brazil in 2012. Her main research area is software engineering, with contributions on software architecture, software design, implementation and evolution, and self-adaptive systems.

Alberto Schaeffer-Filho Alberto E. Schaeffer-Filho holds a Ph.D. in Computer Science (Imperial College London, 2009) and is Associate Professor at Federal University of Rio Grande do Sul (UFRGS), Brazil. From 2009 to 2012 he worked as a research associate at Lancaster University, UK. Dr. Schaeffer-Filho is a CNPq-Brazil Research Fellow and his areas of expertise are network/service management, network virtualization and software-defined networks, and security and resilience of networks. He has authored over 70 papers in leading peer-reviewed journals and conferences related to these topics, and also serves as TPC member for important conferences in these areas, including: CNSM (2019), NetSoft (2019), IFIP/IEEE IM (2019), IEEE/IFIP NOMS (2018) and IEEE INFOCOM CNTCV Workshop (2017). He served as the general chair for SBRC 2019, co-chair for IEEE ICC 2018 CQRM Symposium and demo co-chair for IFIP/IEEE IM 2017. He is also a member of the IEEE.