Cost-efficient Placement and Scaling of 5G Core Network and MEC-enabled Application VNFs

Davit Harutyunyan*†, Rasoul Behravesh†‡, and Nina Slamnik-Kriještorac¶
* Corporate Research, Robert Bosch GmbH, Germany, E-mail: davit.harutyunyan@de.bosch.com
† Smart Network and Services, FBK, Italy, E-mail: rbehravesh@fbk.eu
‡ Department of Electrical, Electronic and Information Engineering, University of Bologna, Italy
¶ University of Antwerp - imec, IDLab, Belgium. E-mail: nina.slamnikkrijestorac@uantwerpen.be

***Abstract*—The adoption of the 5G technology is becoming a must for the mobile network operators (MNOs) in order to remain competitive in the market and efficiently cope with the increase in the mobile data traffic while providing support for a number of futuristic use cases and services. Given the unprecedented demand of mobile data traffic and its variation, it is of paramount importance to dynamically scale the 5G core network functions as well as the applications, both of which are expected to be deployed as virtual network functions (VNFs), in order to avoid over/under-utilization of these VNFs.**

In this work, we study the problem of a joint user association, service functions chain (SFC) placement, and VNF scaling with a particular emphasis on analyzing the trade-offs between the VNF scaling strategies. Specifically, we compare vertical, horizontal, and hybrid VNF scaling strategies by formulating an integer linear programming (ILP) problem that aims at minimizing the service provisioning cost for the MNOs, while satisfying users' data rate requirements. The users' service requests are represented as SFCs composed of end-to-end mobile network components (e.g., gNBs, 5G core network VNFs, and application VNFs). Finally, we devise a heuristic algorithm to tackle the scalability issue of the ILP-based approach.

***Index Terms*—Vertical/horizontal VNF scaling, SFC placement, MEC, Resource Allocation, User Association, 5G Networks.**

I. Introduction

As opposed to the previous generations of mobile network technologies, humans are expected to become the minority of the users of the fifth-generation mobile network technology (5G), which is foreseen to be used by a multitude of objects such as cars, drones, machineries, traffic lights, spurring realization of a number of futuristic use cases (e.g., automotive, healthcare, smart cities, etc.) [1]. The ever-increasing number of novel applications and services in the 5G networks, along with the growing demand for data traffic, presses an urgent need for mobile network operators (MNOs) to adopt 5G technology in order to remain competitive in the market.

The network function virtualization (NFV) technology plays a pivotal role in the realization of the 5G networks [2]–[4], as it decouples the legacy network functions (NFs) from purpose-built hardware and deploys them as platform-independent virtualized network functions (VNFs). For instance, it enables the NFs such as the user plane function (UPF) and session management functions (SMF) in the 5G core service-based architecture (SBA) [5] to be deployed as VNFs, providing unprecedented management flexibility while curtailing both capital expenditure (CapEx) and operational expenses (OpEx). Additionally, NFV provides the opportunity to represent 5G network services and applications as a single VNF or multiple VNFs interconnected in a particular order forming service function chains (SFCs).

978-3-903176-32-4

Multi-access edge computing (MEC) is yet another technology that is expected to be widely adopted in the 5G network in order to satisfy the ultra-low latency requirement of certain applications and services while at the same time alleviating the transport network load [6]. More specifically, in conjunction with NFV, MEC enables both the 5G core VNFs and the application VNFs to be placed at the most appropriate location at the mobile network edge. It should be mentioned that some of the 5G core VNFs, such as the UPF, are tightly coupled with the application VNFs and, therefore, it is of utmost importance to take into account the interplay between these VNFs in the SFC placement. Thus, not only the services but also the UPF should be placed at the MEC servers closer to the users in order to reduce the latency experienced by the users [7], [8]. While MEC servers can be co-located with various network nodes such as gNBs - which are 5G radio nodes providing radio coverage to the users and processing their baseband signals - and 5G core nodes, the closer is the MEC server to the gNB, the less is its computational capacity.

The rapid change in the mobile data traffic demand calls for efficient approaches to dynamically adjust the mobile network's capacity according to the demand. MNOs shall be able to increase/decrease the capacity of both the 5G core network and application VNFs upon the need, ensuring optimal resource utilization, and lowering the service provisioning cost. This is where the vertical, horizontal, and hybrid VNF scaling strategies come into play. While the vertical VNF scaling implies that the existing VNF is resized upon the need, adding/removing computational, memory, or storage resources, in the case of the horizontal VNF scaling, another instance of the same VNF is spawned/terminated. Although horizontal scaling ensures high scalability and reliability of the service, it suffers from increased resource consumption and state migration challenges. On the other hand, while vertical scaling provides higher utilization of resources, thereby creating resource-optimized VNFs, its lower scalability and inability to change the VNF host significantly affect its practical implementation. Since both scaling strategies have their pros and cons, applying only vertical or horizontal scaling strategy cannot perform well in all the scenarios. This is why it is important to consider the so-called hybrid VNF scaling strategy, in which it is possible to perform either vertical or

horizontal VNF scaling depending upon the need. However, it is a non-trivial task to decide which type of VNF scaling to perform since there are a number of parameters (e.g., the VNF type, its resource requirements) to take into account.

After performing VNF scaling, the placement of the VNF is another challenge that requires careful considerations. On the one hand, the interconnections between VNFs composing an SFC must be taken into account in order to make an optimal placement decision. On the other hand, the resource scarcity of the MEC servers at the network edges (e.g., collocated with gNBs) must be considered in order to efficiently utilize the network resources while at the same time satisfying the quality-of-service (QoS) requirement of the requested service.

This paper demonstrates the pros and cons of vertical, horizontal, and hybrid VNF scaling strategies. To this end, we formulate and solve a joint user equipment (UE) association, SFC placement, and VNF scaling problem by leveraging integer linear programming (ILP) techniques, having the objective of minimizing the service provisioning cost while satisfying users' QoS requirements. The UE requests are represented as SFCs that encompass the end-to-end mobile network such as a gNB, 5G core network component and an application with the latter having a specific data rate requirement. We then propose a heuristic algorithm to address the scalability issue of the ILP formulation. The simulations results demonstrate the effectiveness of the proposed hybrid VNF scaling heuristic algorithm in terms of resource utilization of the nodes/VNFs and the availability of the VNFs compared to the ILP-based vertical and horizontal VNF scaling algorithms.

II. Related Work

A. VNF Scaling

A significant research effort has been invested in studying the problem of VNF scaling [9]–[16]. In [9], Sedaghat et al. study the trade-off between cost and performance for vertical and horizontal scaling of Virtual Machines (VM). The study in [10] evaluates the performance of the horizontal, vertical, and hybrid scaling in the cloud environment, concluding that while horizontal scaling has the lowest overhead, the hybrid method offers the highest flexibility. Furthermore, Wang et al. in [11] conclude that under low throughput demand and budget constraint, the VM scale-out operation is preferable for cloud environments, while the VM scale-up operation is more appealing for high throughput demand.

While a vertical auto-scaling algorithm is proposed by Buyakar et al. in [12] for data plane VNFs in the 4G core network to avoid under-utilization of network slices, a control theory-based VNF horizontal scaling method is put forward in [13] for AMF VNF in the 5G core network, studying also load balancing among the AMF instances. A fixed threshold-based vertical scaling approach is introduced by Moghaddassian et al. in [14]. The resource threshold is updated based on the real-time monitoring of the data utilization, while the scaling-down/up decisions are made during an observation period. A queuing theory-based and a mathematical model are proposed in [15] to formulate an optimization problem that aims to minimize the VNF's processing time and link transmission delay. Both vertical and horizontal VNF scaling strategies are separately considered. Tang et al. [16] study the horizontal VNF scaling problem proposing an approach to forecasting the load on the VNFs to scale them on time. Thus, while there are a number of works studying the vertical and horizontal VNF scaling in solitude, there are just a few studies that consider the hybrid VNF scaling strategy. Among the latter ones, our study stands out since, apart from scale-up/down, it also considers scale-in/down operations of a VNF.

B. UE Association

Numerous studies have been conducted also on the UE association in mobile networks [17]–[21]. The problem of user association in heterogeneous mobile networks is formulated by Liu et al. as a Nash bargaining problem [17]. The objective of this work is to minimize data rate utility while balancing the load among base stations and respecting users' requirements. The same problem is formulated and studied as a non-convex non-linear programming problem in [18] to minimize power consumption. On the other hand, Amine et al. [19] study the same problem with a multi-objective approach, proposing a genetic algorithm to reach a near-optimal solution while meeting users' demands. A mobility-aware user association method is proposed in [20] for mmWave mobile networks. The proposed method is able to track the variations in the network topology and channel condition. Authors of [21] study the user association problem in a cache-enabled mobile network, capturing the trade-off between the radio access network and the transport network utilization in 5G networks. As opposed to these studies, our study takes into account the mobility of the UEs, which further intricates the joint UE association, SFC placement, and VNF scaling problem.

C. VNF/SFC Placement

The challenge of VNF/SFC placement has also been thoroughly studied in the state-of-the-art literature [22]–[29]. Alleg et al. [22] propose an optimization model for SFC placement based on mixed-integer quadratically constrained program. The work presented by Zhang et al. [23] considers a hierarchical 5G network composed of edge, core, and cloud servers for hosting VNFs with the objective of maximizing the throughput of the VNFs, given the adverse effects of VNF consolidation in terms of VNF interference. The study in [24] considers a similar architecture and proposes an ILP-based algorithm and a heuristic to minimize the overall network latency. The authors of [26] employ a queuing-based system to solve the problem of VNF placement and CPU allocation to minimize the ratio between the actual and maximum allowed latency. A joint user association, SFC placement, and resource allocation problem is studied in [27] with the goal of minimizing E2E latency, SFC provisioning cost, and VNF migration frequency. Similarly, [28] studies the same problem having the objective of minimizing the number of VNF instances, link utilization, and VNF provisioning cost. The work presented in [29], on the other hand, studies the network slice embedding problem where each slice has a certain QoS requirement and is represented as an SFC that encompasses the radio access and the 5G core networks. However, to the best of our knowledge, our work is the first one that jointly considers the placement of

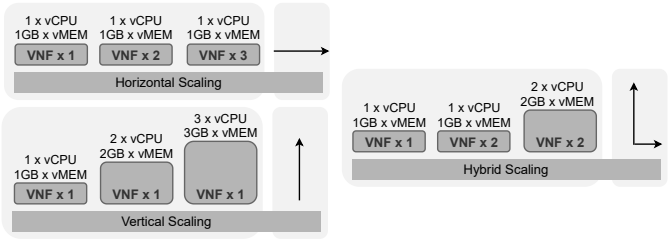


Fig. 1: Horizontal, vertical, and hybrid VNF scaling.

both the 5G core network VNFs and the application VNFs of different types (e.g., CPU/MEM-bound).

III. VNF Scaling Strategies

Since the main focus of our paper is on the VNF scaling problem, we shall first introduce the VNF scaling strategies studied in our work. The top left side of Fig. 1 illustrates the horizontal VNF scaling strategy. It is assumed that an autoscaling group is defined with a minimum and maximum number of VNF instances, and that there is a VNF template (i.e., VNF descriptor) with specific resource requirements that is used to spawn VNF instances. In the case of the scale-out operation, one or multiple instances of the same VNF are deployed with identical resource flavor, using a pre-defined VNF template, while in the case of the scale-in operation, one or more of the available VNF instances are terminated. In the horizontal VNF scaling, it is also assumed that the new VNF instance is preferably deployed on a different node, thus, guaranteeing high availability of the VNF/service, which is one of the prominent advantages of this VNF scaling strategy. Another advantage of the horizontal VNF scaling is its scalability, which is achieved due to small resource footprint of the VNFs.

The bottom left side of Fig. 1 displays the vertical VNF scaling strategy. This type of VNF scaling implies that there is no change in the number of VNFs, while the current VNF is resized adding/curtailing a certain amount of resources (e.g, CPU, MEM, or both) in case of VNF scale up/down operation. In order to guarantee service continuity of the VNF, it is assumed that the VNF is terminated only after deploying its resized instance. The downside of the approach, however, is that it requires availability of resources on the host node even during the scale-down operation of the VNF, as opposes to the scale-in operation in the horizontal VNF scaling strategy. The vaunted benefits of the vertical VNF scaling include enabling the creation of a CPU-optimized or MEM-optimized VNF; that is, if only more CPU is required for a VNF, for example, then a new (i.e., resized) VNF instance could be spawned with more CPU resources, while leaving the MEM resource the same. This is in contrast to the horizontal VNF scaling strategy, which would just instantiate another VNF allocating both CPU and MEM resources even without the need for extra MEM resource. While the vertical VNF scaling is significantly less scalable, it is a better strategy in terms of data consistency than its horizontal counterpart. This is because the resized VNF instances are preferably placed on the same host node exempting the need for the VNF/application state transfer from one node to another in case the considered VNF is stateful.

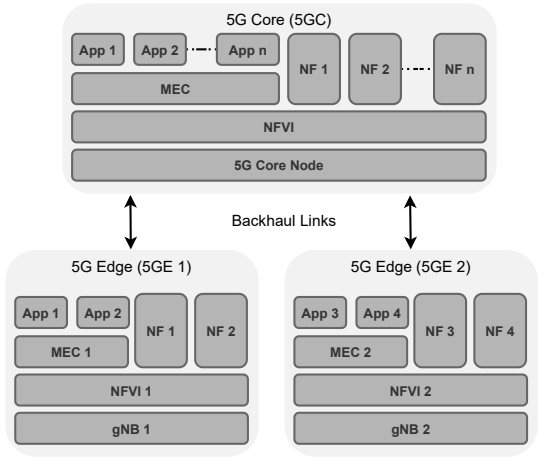


Fig. 2: Physical architecture of the mobile network.

For more details on the characteristics of these VNF scaling strategies, we refer the reader to [30].

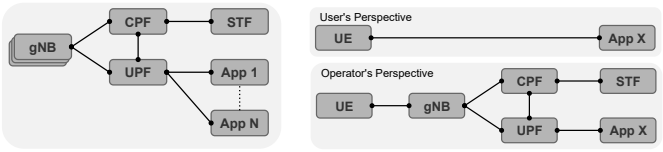
As discussed above, both the vertical and horizontal VNF scaling strategies have pros and cons. Each of these strategies perform better in a specific scenario. Therefore, more benefits can be reaped by using the hybrid VNF scaling strategy shown on the right-hand side of Fig. 1. This strategy incorporates both vertical and horizontal VNF scaling approaches, thereby enabling selection of the most appropriate approach for a VNF while considering a number of parameters such as the VNF type, its resource utilization and the QoS requirements, etc.

IV. Network Model

A. Problem Statement

The physical architecture of the mobile network considered in this study is composed of a 5G core (5GC) and 5G Edges (5GEs), with the latter encompassing a gNB, as shown in Fig. 2. The 5GC is connected to the 5GEs by means of backhaul links. While both the 5GC and the 5GEs have deployed network functions virtualization infrastructure (NFVI), the one of the 5GC possesses significantly more computational (vCPU) and memory (vMEM) resources compared to that of 5GEs. By virtualizing the underlying hardware resources, NFVIs are capable of hosting VNFs [31]. It is assumed that each NFVI has already hosted a UPF and a MEC server as VNFs with the latter being able to host UE requested application VNFs [32]. Apart from the MEC server, the NFVIs can also host 5GC network functions (NFs) such as stateful functions and control plane functions (CPFs) as detailed in the subsequent section, which are also deployed as VNFs.

The mobile network's logical architecture is composed of gNBs, 5GC VNFs, and various application VNFs, as depicted in Fig. 3a. It is worth to mention that, thanks to NFVI and MEC servers, the logical mobile network can be mapped to either only a 5GE node or a composition of 5GE and 5GC nodes in the physical mobile network architecture depicted in Fig. 2. The 5GC VNFs are grouped into stateful functions (STFs), CPFs, and a UPF. While for the sake of simplicity, it is assumed that STFs and CPFs can be deployed as a single VNF, they consist of multiple stateful and control plane functions, respectively. For instance, the STFs in the SBA are



(a) Logical network architecture. (b) UE request.

Fig. 3: Logical mobile network architecture and UE request.

composed of NF repository function (NRF), unstructured data storage function (USDF), and unified data repository (UDR), while the CPFs contain network functions such as access and mobility management function (AMF), and SMF. STFs store UEs' subscription data, dynamic state data, application data, as well as the profiles of different NF instances. CPFs instead handle all the control plane signaling between the NFs and the UEs, performing authentication, session, and mobility management for the UEs. The UPF, on the other hand, is in charge of routing and forwarding the packets received either by the UEs through the gNBs in the uplink direction or by the applications deployed on the MEC servers in the downlink direction. Note that there may be multiple applications running on the MEC server, which can be accessed through the UPF as per ETSI [33]. For more information on 5G SBA and the functionalities of its NFs, we refer the reader to [34].

It is possible to deploy various applications on the mobile network in order to serve the requests of the UEs. While from the UEs' perspective, the application request is quite simple, it is more complex from the MNOs' perspective, as Fig. 3b illustrates in the upper and the lower parts, respectively. This is because in order to set up a network communication for a UE, regardless of the kind of the procedure the UE performs (e.g., attachment, handover, etc.), the UE has to be associated with a gNB, which, in turn, should establish a data plane and a control plane communication with their respective UPF and CPFs, and access UE-related subscription/state information from STFs. Hence, the requests of the UEs are represented as SFCs encompassing the components of the end-to-end mobile network, as illustrated in Fig. 3b.

The optimal placement of these SFCs depends on various factors, including the resource availability, their cost, and QoS requirements of the requested service, such as the date rate and the E2E latency. While the E2E latency demand of an SFC plays a major role in the placement decision of the SFC, it is not considered in this work since the main focus of this study is on the VNF scaling strategy. For a detailed consideration of the SFC placement with an E2E latency requirement, we refer the reader to our previous study [35]. The problem of joint UEs' association, SFC placement, and VNF scaling can be stated as follows:

Given: a 5G mobile network with i) each node (e.g., 5GEs, 5GC) having a certain amount of vCPU and vMEM resources, ii) the transport network with the capacity of the backhaul links, and iii) a number of UEs making application requests with specific data rate demand.

Find: association of the UEs, the placement of the application SFCs, as well as the appropriate VNF scaling, if needed.

Objective: minimize the MNO's service provisioning cost.

Note that the mobile network/infrastructure provider is assumed to be the same entity providing the applications/services implemented by the SFCs. The proposed optimization approach, however, can be easily adapted to consider also the case in which these entities are different.

TABLE I: Mobile network parameters

Parameter	Description
G_{net}	Mobile network graph.
N_{net}	Set of nodes/DCs in G_{net}.
N_{5gc}	The 5GC in G_{net}.
N_{5ge}	Set of 5GEs in G_{net}.
N_{gnb}	Set of gNBs, one per 5GE in G_{net}.
$N_{upf,cfp}$	Set of user plane and control plane function VNFs.
$N_{stf,app}$	Set of stateful function (STF) and application VNFs.
N_{vnf}	Set of all VNFs, $N_{vnf} = N_{upf} \cup N_{cpf} \cup N_{stf} \cup N_{app}$.
N^v_{flv}	Flavours of the VNF v, $N^v_{flv} = N^v_{v-flv} \cup N^v_{h-flv}$.
E_{net}	Set of backhaul links in G_{net}.
$\omega_{prb}(g)$	PRB resources of gNB g n.
$\omega_{c,m}(n)$	Computational and memory resources of the node n.
$\omega^{vnf}_{c,m}(f)$	Computational and memory resources of the flavour $f \in N^v_{flv}$ of the VNF $v \in N_{vnf}$.
$\omega^{upf,app}_t(f)$	Throughput of flavour $f \in N^{upf,app}_{flv}$ of UPF and application VNF.
$\omega^{cpf}_e(f)$	Number of events supported by CPF flavour $f \in N^{cpf}_{flv}$.
$\omega^{stf}_q(f)$	Number of queries supported by STF flavour $f \in N^{stf}_{flv}$.
$\omega_b(e^{nm})$	Capacity of the link $e^{nm} \in E_{net}$.
$loc(n)$	Geographical location of the node $n \in N_{net}$.
$\delta(g)$	Coverage radius of the gNB $g \in N_{gnb}$ (in meters).

B. Mobile Network Model

Let $G_{net} = (N_{net}, E_{net})$ be an *undirected* graph modelling the physical architecture of the mobile network, where $N_{net} = N_{5ge} \cup N_{5gc}$ is the union of the set of 5GEs and a 5GC. There is a one-to-one mapping between a 5GE and a gNB. Each network node $n \in N_{net}$ has a certain amount of vCPU $\omega_c(n)$ and vMEM $\omega_m(n)$ resource. Additionally, each node $n \in N_{net}$ is associated with a geographic location $loc(n)$, as x, y coordinates while each gNB $g \in N_{gnb}$ is also associated with a coverage radius of $\delta(g)$, in meters. E_{net} is the set of backhaul links between the 5GEs and the 5GC. An edge $e^{nm} \in E_{net}$ exists if and only if there is a connection between $n, m \in N_{net}$. A weight $\omega_b(e^{nm})$ is assigned to each edge $e^{nm} \in E_{net} : \omega_b(e^{nm}) \in \mathbb{N}^+$ denoting its capacity, in Gbps.

All the nodes (i.e., 5GEs and 5GC) are able to host both 5GC network VNFs as well as application VNFs on their MEC server, which, in turn, is deployed as a VNF, however, is not a subject of scaling in our study. Each VNF $v \in N_{vnf}$ may have multiple flavours $N^v_{flv} = N^v_{v-flv} \cup N^v_{h-flv}$, each representing either another instance of the same VNF with identical amount of resources, referred to as horizontal flavour $f \in N^v_{h-flv}$, or a resized version of that VNF with more/less vCPU, and/or vMEM resources, referred to as vertical flavour $f \in N^v_{v-flv}$. Each flavour $f \in N^v_{flv}$ of the VNF $v \in N_{vnf}$, where $N_{vnf} = N_{upf} \cup N_{cpf} \cup N_{stf} \cup N_{app}$, has a certain amount of vCPU and vMEM $\omega^{vnf}_{c,m}(f)$ resources. These resources for the UPF and application VNFs (N_{upf} and N_{app}) are translated into a maximum amount of supportable traffic $\omega^{upf,app}_t(f)$, while for STF and CPF VNFs (N_{stf} and N_{cpf}), they are

TABLE II: UE request parameters

Parameter	Description
G_{req}	UE request graph.
N_{ue}	Set of UEs in G_{req}.
N^{ue}_{vnf}	Set of VNFs in G_{req}.
$N^{ue}_{app}(u)$	Application VNF requested by UE $u \in N_{ue}$.
$\omega^{ue}_{d}(u)$	Data rate requested by UE $u \in N_{ue}$.
$\omega^{ue}_{prb}(u)$	Number of PRBs needed to satisfy $\omega^{ue}_{d}(u)$.
$\omega^{ue}_{e}(u)$	Number of events generated by UE $u \in N_{ue}$.
$\omega^{ue}_{q}(u)$	Number of queries generated by UE $u \in N_{ue}$.
$E_{req}, E_{req}(u)$	Set of all virtual links, and that of the UE $u \in N_{ue}$.

TABLE III: Decision variables.

Variable	Description
χ^{u}_{g}	Indicates if UE $u \in N_{ue}$ is associated with gNB $g \in N_{gnb}$.
$\chi^{v}_{f,n}$	Indicates if VNF $\hat{v} \in N^{ue}_{vnf}(u)$ requested by UE u is mapped to the flavour $f \in N^{v}_{flv}$ of the same VNF on node $n \in N_{net}$.
$\chi^{v}_{f,n}$	Indicates if the flavour $f \in N^{v}_{flv}$ of the VNF $v \in N_{vnf}$ on node $n \in N_{net}$ has been used.

expressed, respectively, in terms of a maximum number of supportable queries $\omega^{stf}_{q}(f)$ and signalling events $\omega^{cpf}_{e}(f)$. Table I summarizes the mobile network parameters.

C. UE Request Model

Both voice UEs and data UEs are considered. While the former is engaged in a voice call and, therefore, requires a call communication setup generating control plane events and queries towards, respectively, CPFs and STFs, the latter is using an application, which apart from communication with CPFs and STFs, requires also a data plane communication with the application service. All UE requests are modelled as *directed* graphs $G_{req} = (N_{req}, E_{req})$ where $N_{req} = N_{ue} \cup N^{ue}_{vnf}$ is the union of the set of UEs and the set of VNFs (i.e., UPFs, CPFs, STFs, applications) that each UE has to have a connection with. E_{req} is the set of virtual links between UEs and their respective VNFs. Each data UE $u \in N_{ue}$ requires a certain amount of data rate $\omega^{ue}_{d}(u)$ for its application. Additionally, it is assumed that the UE communication setup process (e.g., during the initial UE to a gNB association, during the UE handover, etc.) for each UE generates a certain fixed amount of signaling events $\omega^{ue}_{e}(u)$ and queries $\omega^{ue}_{q}(u)$, which have to be handled by the CPFs and STFs, respectively.

V. Scaling Algorithms

Upon receiving UE requests, the MNO shall i) associate the UE with a gNB, ii) either place new VNFs on the computing nodes or use the already existing VNFs with/without scaling them, and iii) allocate enough computing resources to accommodate the request, if necessary. The goal is to satisfy the SFC requirements of the UE requests while at the same time making sure that the substrate network resources are used in an efficient manner. The SFC placement is modeled as a virtual network embedding (VNE) problem, which is *NP–hard* and has been studied extensively in the literature [36], [37]. The embedding process consists of two parts: the node embedding and the link embedding. In the node embedding, each virtual node in the request (e.g., CPF, STF) is mapped to a substrate node (e.g., 5GE, 5GC). In the link embedding instead, each virtual link is mapped to a single substrate path. In both cases, the constraints of the nodes and links must be satisfied for an SFC mapping solution to be valid.

A. ILP Formulation

Before formulating the ILP model, for each UE, we first need to find the set of gNBs that provide coverage. Considering the location $loc(u)$ of the UE $u \in N_{ue}$ along with the location $loc(g)$ and the coverage radius $\delta(g)$ of gNB $g \in N_{gnb}$, the set of candidate gNBs $\Omega(u)$ for the UE u can be defined as follows:

$$\Omega(u) = \{g \in N_{gnb} | dis(loc(g), loc(u)) \leq \delta(g)\} \quad (1)$$

Formula (2) represents the objective functions considered in this ILP formulation. The first two arguments in (2) calculate the VNF deployment cost for, respectively, vertical and horizontal VNF scaling cases. The third argument takes into account the backhaul link usage cost, while the last one considers the Physical Resource Block (PRB) usage cost at the gNBs. ξ_c, ξ_m, ξ_b and ξ_p represent the cost of, respectively, a single vCPU core, 1Mb vMEM, 1Mbps backhaul bandwidth, and one PRB. Table III shows all used binary variables.

Specifically, three objective functions are considered in this ILP formulation. While they pursue the same goal of minimizing the service provisioning cost for the MNO, they differ in terms of the used VNF scaling strategy, which is enforced by binary coefficients Λ_v and Λ_h. More specifically, if $\Lambda_v = 1, \Lambda_h = 0$ in formula (2) then only vertical VNF scaling is considered in this objective. If $\Lambda_v = 0, \Lambda_h = 1$ then only horizontal VNF scaling is considered in this objective. Finally, if $\Lambda_v = 1, \Lambda_h = 1$ then both the vertical and horizontal VNF scaling strategies are considered in the objective function, referred to as hybrid scaling, enabling the algorithm to pick the most appropriate VNF scaling strategy for a specific VNF based on a number of parameters such as VNF type and resource requirements.

$$\begin{aligned} &Minimize: \\ &\Lambda_v \sum_{v \in N_{vnf}} \sum_{f \in N^{v}_{v-flv}} \sum_{n \in N_{net}} (\xi_c \omega^{vnf}_{c}(f) + \xi_m \omega^{vnf}_{m}(f)) \chi^{v}_{f,n} + \\ &\Lambda_h \sum_{v \in N^{vnf}} \sum_{f \in N^{v}_{h-flv}} \sum_{n \in N_{net}} (\xi_c \omega^{vnf}_{c}(f) + \xi_m \omega^{vnf}_{m}(f)) \chi^{v}_{f,n} + \\ &\sum_{u \in N_{ue}} \sum_{\hat{e} \in E_{req(u)}} \sum_{e \in E_{net}} \xi_b \omega^{ue}_{d}(u) \chi^{\hat{e}}_{e} + \sum_{u \in N_{ue}} \sum_{g \in N_{gnb}} \xi_p \omega^{ue}_{prb}(u) \chi^{u}_{g} \end{aligned} \quad (2)$$

Regardless of the considered objective function, all the following constraints have to be satisfied in order for an SFC placement solution to be valid. In the considered scenario, each UE $u \in N_{ue}$ has to be associated to only one gNB $g \in N_{gnb}$ (Constraint (3)), which belongs to the set of candidate gNBs $\Omega(u)$ of that UE (Constraint (4)) and has to have sufficient amount of PRBs in order to satisfy the data rate demand of all the UEs that are associated to that gNB (Constraint (5)).

$$\forall u \in N_{ue} : \sum_{g \in N_{gnb}} \chi^{u}_{g} = 1 \quad (3)$$

$$\forall u \in N_{ue} : \sum_{g \in N_{gnb} \setminus \Omega(u)} \chi_g^u = 0 \tag{4}$$

$$\forall g \in N_{gnb} : \sum_{u \in N_{ue}} \omega_{prb}^{ue}(u) \chi_g^u \leq \omega_{prb}(g) \tag{5}$$

Since each VNF flavour $f \in N_{flv}^v$ has a certain amount of vCPU and vMEM resource requirement $\omega_{c,m}^{vnf}(f)$, each substrate network node $n \in N_{net}$, be it a 5GE or a 5GC, can host flavours of different VNF types (e.g., UPF, CPF, STF, application) as long as it has sufficient amount of vCPU and vMEM resources to host the VNF flavour (Constraint (6)).

$$\forall n \in N_{net} : \sum_{v \in N_{vnf}} \sum_{f \in N_{flv}^v} \omega_{c,m}^{vnf}(f) \chi_{f,n}^v \leq \omega_{c,m}(n) \tag{6}$$

The VNF flavours can serve the UEs as long as they have enough capacity (see Constraints (7), (8) and (9)). The VNF types are characterized by different sorts of resources. Specifically, while the UPF and application VNFs are characterized by a maximum amount of supportable traffic enforced by Constraint (7), the CPF VNF and the STF VNF are characterized, respectively, by a maximum number of supported control plane events (see Constraint (8)) and queries (see Constraint (9)).

$$\forall v \in N_{upf,app}, \hat{v} \in N_{vnf(v)}^{ue}, \forall f \in N_{flv}^v, \forall n \in N_{net} : \sum_{u \in N_{ue}} \omega_d^{ue}(u) \chi_{f,n}^{\hat{v}} \leq \omega_t^v(f) \tag{7}$$

$$\forall v \in N_{cpf}, \hat{v} \in N_{vnf(v)}^{ue}, \forall f \in N_{flv}^v, \forall n \in N_{net} : \sum_{u \in N_{ue}} \omega_e^{ue}(u) \chi_{f,n}^{\hat{v}} \leq \omega_e^v(f) \tag{8}$$

$$\forall v \in N_{stf}, \hat{v} \in N_{vnf(v)}^{ue}, \forall f \in N_{flv}^v, \forall n \in N_{net} : \sum_{u \in N_{ue}} \omega_q^{ue}(u) \chi_{f,n}^{\hat{v}} \leq \omega_q^v(f) \tag{9}$$

Constraint (10) makes sure that each UE is connected to a single flavour of each VNF type that composes the UE's SFC request, while Constraint (11) guarantees that a VNF flavour is used if at least one UE uses it, where μ is a big number.

$$\forall u \in N_{ue}, \forall \hat{v} \in N_{vnf}^{ue}(u) : \sum_{n \in N_{net}} \sum_{f \in N_{flv}^{\hat{v}}} \chi_{f,n}^{\hat{v}} = 1 \tag{10}$$

$$\forall \hat{v} \in N_{vnf}^{ue}, \forall f \in N_{flv}^{\hat{v}}, \forall n \in N_{net} : \sum_{u \in N_{ue}} \chi_{f,n}^{\hat{v}} - \mu \chi_{f,n}^v \leq 0 \tag{11}$$

The backhaul link capacity constraint is handled by Constraint (12), which ensures that the virtual links can be mapped on a substrate backhaul link if the one has sufficient bandwidth to support the data rate demand of the virtual links. Lastly, Constraint (13) enforces for each virtual link to be a continuous path established between the gNB hosting the UE and the nodes hosting the VNFs of the SFC requested by the UE. $E_{net}^{\star i}$ is the set of the links that originate from any node and directly arrive at the node $i \in N_{net}$, while $E_{net}^{i\star}$ is the set of links that originates from the node i and arrive at any node directly connected to i.

$$\forall e \in E_{net} : \sum_{u \in N_{ue}} \sum_{\hat{e} \in E_{req(u)}} \omega_d^{ue}(\hat{e}) \chi_e^{\hat{e}} \leq \omega_b(e^{nm}) \tag{12}$$

$$\sum_{e \in E_{net}^{\star i}} \chi_e^{e^{n,m}} - \sum_{e \in E_{net}^{i\star}} \chi_e^{e^{n,m}} = \begin{cases} -1 & \text{if } i = n \\ 1 & \text{if } i = m \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

$$\forall i \in N_{net}, \quad \forall e^{n,m} \in E_{req}$$

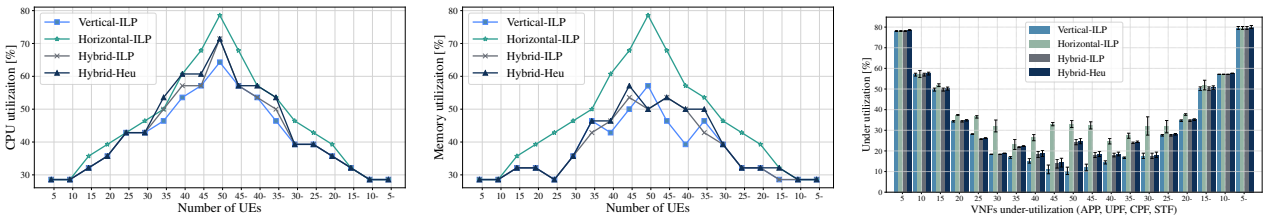
B. Heuristic

The ILP formulation becomes computationally intractable as the network's size increases (e.g., the number of gNBs, VNFs, and UEs). For example, it takes a day on an Intel Core i7 laptop (3.0 GHz CPU, 16 Gb RAM) using Gurobi solver [38] to associate and serve 50 UEs making service requests, each composed of 4 VNFs in a network composed of 4 gNBs and a core. To tackle the scalability issue of the ILP formulation, we propose a heuristic (the pseudo code is not shown due to space limitation) able to find near-optimal solutions for all the requests in a considerably shorter time.

The proposed heuristic pursues the hybrid VNF scaling objective of the ILP formulation and consists of three phases. The algorithm iterates over all the gNBs to find candidate gNBs for each UE in the first phase. A gNB is considered as a candidate if the UE is under the coverage radius of the gNB, and the gNB has a sufficient amount of PRBs to satisfy the UE's data rate demand.

In the second phase, the algorithm tries to serve the UEs from the existing VNFs in the network. The algorithm begins to serve UEs in sequence by looping over all the UEs, the candidate gNBs of the UE, and all the computing nodes to find if an instance of the UE's requested VNFs exists on the node. If yes, then the cost of serving that VNF instance will be computed. The cost encompasses the cost of PRB resources to associate the UE with the candidate gNB and the link resources that are needed to make a continuous path from the UE to the VNF instance. It is worth noting that this phase does not take into account the cost of using CPU and memory resources since the VNF instance already exists on the node, and there is no need to allocate computing resources to embed the VNF. After computing the cost for each possible solution (i.e., VNF, nodes, gNB) and finding a solution with a minimum cost, the VNF instance will be allocated to the UE, the UE will be associated with a gNB, and a path will be established from the gNB that the UE is associated with to the node hosting the VNF. This is followed by updating the network resources.

The third phase of the algorithm attempts to accommodate those UE requests for which there was no preexisting candidate VNF in the second phase. Thus in this phase, the algorithm tries to instantiate a new VNF instance of the requested service. Like the ILP, a weighting factor is considered for computing the cost of all the different flavours of each VNFs. In this regard, the algorithm sorts the solutions based on the cost in ascending order and loops over all the solutions until reaching a case that leads to minimum cost. Unlike the second phase, the solution cost in this phase encompasses the computing resources cost, link cost, and PRB cost. After finding a flavour of the VNF instance that complies with the node, link, and PRB resource demand, the VNF flavour will be embedded on the node, and the resources will be allocated then updated. Regarding the time complexity of the proposed



(a) CPU utilization. (b) Memory utilization. (c) VNF under-utilization.

Fig. 4: CPU and memory utilization of the nodes along with the under-utilization of VNFs for the VNF scaling algorithms.

heuristic algorithm, it is in the order of $O(mnpk)$, where m, n, p and k represent, respectively, the number of UEs, nodes, VNFs, VNF flavours.

VI. Evaluation

A. Simulation Environment

A mobile network composed of 5 nodes is considered in this work, out of which one is the core, and the rest are gNBs. The gNBs are connected to the core node through 10 Gbps backhaul links. Both the core node and the gNBs have collocated MEC servers that possess, respectively, 12 and 4 CPU cores and 12 and 4-gigabyte memory, and are endowed with virtualization capability. Each CPU core is assumed to have 1.5 GHz clock rate. We assume four VNF categories, UPF, CPF, STF, and applications, with the last being in 5 types differentiated by their provided services and resource boundness (i.e., CPU-bound, memory-bound, or CPU-memory-bound). Each VNF instance can be shared among multiple UEs as long as it has sufficient capacity. Moreover, each VNF is available in multiple flavors, which defines the combination of CPU and memory resources allocated to that VNF. A VNF instance can have at least/most one/three CPU core(s) and one/three GB of memory.

Two different types of mobile UEs are considered in this work, as already introduced in Section V. The first type of UEs are data UEs that make data requests and use application VNFs. The second type of UEs are voice UEs that do not ask for application but use voice services in the network, generating control messages. The purpose of considering voice UEs in the system is to show the impact of having different request types with diverse requirements and increasing the load on STF and CPF components to trigger scaling operation. The UE requests arrive sequentially in batches, each composed of 5 requests. It is assumed that with the arrival of a new data UE batch, there are four voice UE batches and that the UEs from the previous batches change their locations by moving in random direction with speed selected from the set $\{5, 25, 50\}$km/h, mimicking pedestrians, cyclist, and cars, while still keeping their data rate requirements. We consider up to 10 batches of data UE (50 data UEs in total) intending to trigger both scale-up and scale-out operations. After 50 data UEs, we gradually decrease the number of data UEs by 5 to trigger the scale-down and scale-in operations, if necessary, releasing the allocated resources.

As already mentioned, the capacity of CPF and STF VNFs is characterized, respectively, in terms of a maximum number of supportable events and queries, while UPF and application VNFs are characterized in terms of throughput. All of these metrics are derived from the CPU and memory resource of the VNF. Specifically, it is assumed that the throughput, event capacity, and query capacity of, respectively, UPF, CPF, and STF are 80% dependent on the CPU and 20% on the memory. The capacity of CPF and STF VNFs has been computed following the approach in [39]. The CPU contribution in the overall capacity of a VNF is computed as the number of cores multiplied by each core's clock rate divided by the number of clocks required to process one bit of data (considered 10 in our scenario). Besides, if an application VNF is CPU-bound/MEM-bound then the throughput mostly depends on the CPU/MEM. In the case of CPU-memory-bound application VNF instead, the throughput is equally dependent on the CPU and memory of that VNF.

B. Simulation Results

The simulations are carried out in Python using Gurobi mathematical optimization solver [38]. The reported results the average of 5 simulations with 95% of confidence intervals.

Resource utilization. Figure 4 illustrates the CPU and the memory utilization of the nodes (for a single simulation run) together with the under-utilization of the VNFs as a function of the number of UE requests for both ILP-based and heuristic VNF scaling algorithms. As expected, the vertical and the horizontal VNF scaling strategies achieve, respectively, the lowest and the highest CPU and memory utilization at the computing nodes (e.g., 5GEs, 5GC) as shown in Fig. 4a and Fig. 4b. This is because the horizontal VNF scaling strategy instantiates more VNFs of the same type allocating both CPU and memory resources from the host node even if the scaling is triggered due to the lack of only CPU or memory resource. Conversely, the vertical scaling strategy, thanks to its ability to resize the VNFs according to the need of having more/less CPU, or memory, or both, uses the node resources more efficiently while requiring the least amount of CPU and memory. For the hybrid VNF scaling strategy, we observe that the ILP-based and heuristic algorithms' performance resembles, and their CPU and memory utilization in most cases lies in between the ones achieved by the vertical and horizontal scaling approaches. This is justified by the fact that in this case, depending on the need, both vertical and horizontal VNF scalings are performed, as shown in Fig. 5a.

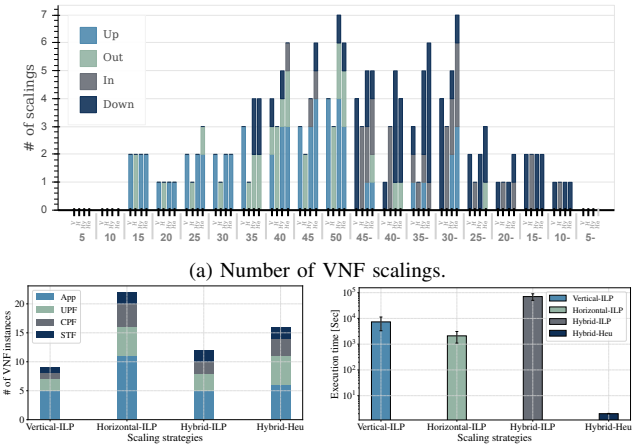
The total under-utilization of the VNFs is computed based on the usage of throughput for UPF and application VNFs, queries for the STF VNFs, and control plane event for the CPF VNFs derived from the CPU and memory of the VNFs. As expected, it reduces with the increase in the number of UE requests (see Fig. 4c). While the horizontal VNF scaling

strategy consumes the highest amount of CPU and memory of the nodes to instantiate VNFs, those VNFs and, therefore, those resources are not used efficiently, leading to the highest total VNF under-utilization. As opposed to the horizontal VNF scaling, the vertical VNF scaling strategy demonstrates the lowest total VNF under-utilization, leading to the most optimal utilization of the VNFs. As for the performance of the hybrid VNF scaling strategies, it is very similar to that of the vertical scaling with a slightly higher VNF under-utilization. It can also be observed that the difference between the VNF under-utilization achieved by the scalings strategies is more evident when the number of UE requests is high. This is because the more is the UEs, the more is the traffic demand, and, therefore, the more are the number of VNF scalings.

Number of VNF scalings. The number of different types of VNF scalings (e.g., scale-up, down, in, out) for the considered ILP-based and heuristic VNF scaling algorithms for varying numbers of UEs is shown in Fig. 5a. As expected, more VNF scaling operations are induced when the number of UEs increases in the network. We can observe that mostly VNF scale-up and scale-out operations perform when the number of UEs increases, with most of the scale-ups/outs being triggered in the case of vertical/horizontal VNF scaling. On the other hand, when the UEs start leaving the network, VNF scale-up and scale-out operations are more dominant. It is worth mentioning that in some rare cases, VNF scale-down and scale-in operations are triggered even if the number of UEs increase in the network, while sometimes VNF scale-up and scale-out are performed when the UEs leave the network. This is due to the ability of the proposed algorithms to perform a customized VNF scaling. For instance, when the scaling-up of a VNF is needed, it might be more efficient to increase the CPU resource and decrease the allocated memory in order to meet the request demand and, at the same time, minimize the provisioning cost and vice versa.

Number of VNF instances. Figure 5b illustrates the average number of VNF instances for all VNF categories for the considered scaling strategies after embedding 50 UE requests. It can be observed that in both the ILP-based and heuristic VNF scaling strategies, there are a way more application VNFs than the other VNF categories, and among the application VNFs, the highest number of VNFs are instantiated by the horizontal VNF scaling, as expected. As for the UPF, CPF, and STF VNFs, there are fewer instances of them due to the fact that these VNFs have much higher capacity, resulting in less frequent scalings. Naturally, the total number of VNF instances of the hybrid VNF scaling strategies lies in between the ones of the vertical and horizontal strategies due to being able to perform both vertical and horizontal VNF scaling.

Execution time. The main motivation for proposing the heuristic algorithm for the hybrid VNF scaling strategy is to address the scalability issue of the ILP-based algorithms. Figure 5 shows the average execution time for all the algorithms for associating 50 UE to the network, embedding the SFC requests, and performing VNF scaling. It can be observed that the execution time of the heuristic algorithm is at least three orders of magnitude less than that of the ILP-based algorithms, making it applicable in more practical scenarios and more suitable for various 5G use cases. Thus,



(a) Number of VNF scalings.

(b) Number of VNF instances.

(c) Execution time.

Fig. 5: Number of different VNF instances, their scaling types, and the execution time for the VNF scaling algorithms.

the heuristic algorithm is much more scalable compared to ILP-based algorithms. Nonetheless, this comes at the expense of sub-optimal mapping solutions, leading to a slightly lower performance compared to its ILP-based counterpart.

Since the proposed algorithms as per their nature are supposed to be run in a centralized location, they are more suitable to be applied to small networks such as non-public networks. MNOs' large networks on the other hand can avail the proposed algorithms by dividing the network into smaller clusters and running them in a centralized location such as on anchor nodes within each cluster.

VII. Conclusions

In this work, we studied a joint UE association, SFC placement, VNF scaling problem in the scenario of an end-to-end 5G network employing ILP and heuristic algorithms. Specifically, vertical, horizontal, and hybrid VNF scaling strategies have been compared, and their trade-offs are analyzed. We demonstrated that while the vertical VNF scaling is the most efficient strategy in utilizing the resources of the nodes and VNFs, it does not provide high availability to those VNFs due to their fewer instances compared to the horizontal VNF scaling strategy. However, the high availability of the horizontal VNF scaling strategy came at the expense of high CPU and memory usage of the nodes and high VNF under-utilization, making it inefficient from the resource utilization perspective. The hybrid VNF scaling strategy, on the other hand, exhibited a better compromise between the high availability of the VNFs and the resource utilization of the nodes and the VNFs. In the future work, we intend to consider also the states of the VNFs in the scaling decision, and the through a proof-of-concept implementation, empirically evaluate the trade-offs between the presented VNF scaling strategies in a 5G network testbed.

Acknowledgments

This work has been performed in the framework of the European Union's Horizon 2020 project 5G-CARMEN co-funded by the EU under grant agreements No. 825012.

REFERENCES

[1] G. P. A. W. Group, "View on 5g architecture," *White Paper, July*, 2016.

[2] ETSI, "Gs nfv 002 v1.2.1," *Network Functions Virtualisation (NFV), Architectural Framework*, 2014.

[3] G. ETSI, "Nfv 001 v1.2.1," *Network Functions Virtualisation (NFV), Use Cases*, 2017.

[4] N. Slamnik-Kriještorac, H. Kremo, M. Ruffini, and J. M. Marquez-Barja, "Sharing Distributed and Heterogeneous Resources toward End-to-End 5G Networks: A Comprehensive Survey and a Taxonomy," *IEEE Communications Surveys Tutorials*, vol. 22, no. 3, pp. 1592–1628, 2020.

[5] E. TS, "123 501 v15.9.0," *5G, System architecture for the 5G System (5GS)*, 2020.

[6] ETSI, "Gr mec 017 v1.1.1," *Mobile Edge Computing (MEC), Deployment of Mobile Edge Computing in an NFV environment*, 2018.

[7] U. Fattore. (2018) Modelling the 5g core for low latency scenarios. [Online]. Available: https://itnspotlight.com/modelling-the-5g-core-for-low-latency-scenarios/

[8] F. Giust, G. Verin, K. Antevski, J. Chou, Y. Fang, W. Featherstone, F. Fontes, D. Frydman, A. Li, A. Manzalini *et al.*, "Mec deployments in 4g and evolution towards 5g," *ETSI White Paper*, vol. 24, pp. 1–24, 2018.

[9] M. Sedaghat, F. Hernandez-Rodriguez, and E. Elmroth, "A virtual machine re-packing approach to the horizontal vs. vertical elasticity trade-off for cloud autoscaling," in *Proceedings of the 2013 ACM Cloud and Autonomic Computing Conference*, 2013, pp. 1–10.

[10] K. Hwang, Y. Shi, and X. Bai, "Scale-out vs. scale-up techniques for cloud performance and productivity," in *2014 IEEE 6th International Conference on Cloud Computing Technology and Science*. IEEE, 2014, pp. 763–768.

[11] W. Wang, L. Xu, and I. Gupta, "Scale up vs. scale out in cloud storage and graph processing systems," in *2015 IEEE International Conference on Cloud Engineering*. IEEE, 2015, pp. 428–433.

[12] T. V. K. Buyakar, A. K. Rangisetti, A. A. Franklin, and B. R. Tamma, "Auto scaling of data plane vnfs in 5g networks," in *2017 13th International Conference on Network and Service Management (CNSM)*. IEEE, 2017, pp. 1–4.

[13] I. Alawe, Y. Hadjadj-Aoul, A. Ksentini, P. Bertin, and D. Darche, "On the scalability of 5g core network: the amf case," in *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2018, pp. 1–6.

[14] M. Moghaddassian, H. Bannazadeh, and A. Leon-Garcia, "Adaptive auto-scaling for virtual resources in software-defined infrastructure," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 2017, pp. 548–551.

[15] R. Gouareb, V. Friderikos, and A.-H. Aghvami, "Virtual network functions routing and placement for edge cloud latency minimization," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 10, pp. 2346–2357, 2018.

[16] H. Tang, D. Zhou, and D. Chen, "Dynamic network function instance scaling based on traffic forecasting and vnf placement in operator data centers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 3, pp. 530–543, 2018.

[17] D. Liu, Y. Chen, K. K. Chai, and T. Zhang, "Nash bargaining solution based user association optimization in HetNets," in *Proc. of IEEE CCNC*, Las Vegas, NV, USA, 2014.

[18] Y. Lei, G. Zhu, C. Shen, Y. Xu, and X. Zhang, "Delay-aware user association and power control for 5G heterogeneous network," *Mobile Networks and Applications*, vol. 24, pp. 1–13, 2018.

[19] M. Amine, A. Walid, A. Kobbane, and J. Ben-Othman, "New user association scheme based on multi-objective optimization for 5g ultra-dense multi-rat hetnets," in *Proc. of IEEE ICC*, Kansas City, MO, USA, 2018.

[20] A. S. Cacciapuoti, "Mobility-aware user association for 5G mmWave networks," *IEEE Access*, vol. 5, pp. 21 497–21 507, 2017.

[21] D. Harutyunyan, A. Bradai, and R. Riggio, "Trade-offs in Cache-enabled Mobile Networks," in *Proc. of IEEE CNSM*, Rome, Italy, 2018.

[22] A. Alleg, T. Ahmed, M. Mosbah, R. Riggio, and R. Boutaba, "Delay-aware VNF placement and chaining based on a flexible resource allocation approach," in *Proc. of IEEE CNSM*, Tokyo, Japan, 2017.

[23] Q. Zhang, F. Liu, and C. Zeng, "Adaptive interference-aware VNF placement for service-customized 5G network slices," in *Proc. of IEEE INFOCOM*, Paris, France, 2019.

[24] Y. Bi, C. Colman-Meixner, R. Wang, F. Meng, R. Nejabati, and D. Simeonidou, "Resource allocation for ultra-low latency virtual network services in hierarchical 5G network," in *Proc. of IEEE ICC*, Shanghai, China, 2019.

[25] S. Yang, F. Li, R. Yahyapour, and X. Fu, "Delay-sensitive and availability-aware virtual network function scheduling for NFV," *IEEE Transactions on Services Computing*, 2019.

[26] S. Agarwal, F. Malandrino, C.-F. Chiasserini, and S. De, "Joint VNF placement and CPU allocation in 5G," in *Proc. of IEEE INFOCOM*, Honolulu, HI, USA, 2018.

[27] D. Harutyunyan, N. Shahriar, R. Boutaba, and R. Riggio, "Latency-aware service function chain placement in 5g mobile networks," in *Proc. of IEEE NetSoft*, 2019.

[28] R. Behravesh, E. Coronado, D. Harutyunyan, and R. Riggio, "Joint user association and vnf placement for latency sensitive applications in 5g networks," in *Proc. of IEEE CloudNet*, 2019.

[29] D. Harutyunyan, R. Fedrizzi, N. Shahriar, R. Boutaba, and R. Riggio, "Orchestrating end-to-end slices in 5g networks," in *2019 15th International Conference on Network and Service Management (CNSM)*. IEEE, 2019, pp. 1–9.

[30] A. N. Toosi, J. Son, Q. Chi, and R. Buyya, "Elasticsfc: Auto-scaling techniques for elastic service function chaining in network functions virtualization-based clouds," *Journal of Systems and Software*, vol. 152, pp. 108–119, 2019.

[31] N. ETSI, "Gs nfv 002-v1. 1.1-network function virtualisation (nfv)-architectural framework," *publishing October*, 2013.

[32] "Multi-access edge computing (mec); framework and reference architecture," *ETSI Group Specification 003*, 2019.

[33] S. Kekki, W. Featherstone, Y. Fang, P. Kuure, A. Li, A. Ranjan, D. Purkayastha, F. Jiangping, D. Frydman, G. Verin *et al.*, "Mec in 5g networks," *ETSI white paper*, 2018.

[34] "3GPP technical specification group services and system aspects; system architecture for the 5G system," 2019.

[35] D. Harutyunyan, N. Shahriar, R. Boutaba, and R. Riggio, "Latency and mobility-aware service function chain placement in 5g networks," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2020.

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

[37] M. Chowdhury, M. R. Rahman, and R. Boutaba, "ViNEYard: Virtual network embedding algorithms with coordinated node and link mapping," *IEEE/ACM Transactions on Networking*, vol. 20, no. 1, pp. 206 –219, February 2012.

[38] "Gurobi mathematical optimization solver." [Online]. Available: https://www.gurobi.com/

[39] F. Z. Yousaf, P. Loureiro, F. Zdarsky, T. Taleb, and M. Liebsch, "Cost analysis of initial deployment strategies for virtualized mobile core network functions," *IEEE Communications Magazine*, vol. 53, no. 12, pp. 60–66, 2015.