VNF and CNF Placement in 5G: Recent Advances and Future Trends

Wissal Attaoui, *Member, IEEE*, Essaid Sabir, *Senior Member, IEEE*, Halima Elbiaze, *Senior Member, IEEE*, and Mohsen Guizani, *Fellow, IEEE*

***Abstract*—With the growing demand for openness, scalability, and granularity, mobile network function virtualization (NFV) has emerged as a key enabler for the most of mobile network operators. NFV decouples network functions from hardware devices. This decoupling allows network services, called Virtualized Network Functions (VNFs), to be hosted on commodity hardware which simplifies and enhances service deployment and management for providers, improves flexibility, and leads to efficient and scalable resource usage, and lower costs. The proper placement of VNFs in the hosting infrastructures is one of the main technical challenges. This placement significantly influences the network's performance, reliability, and operating costs. The VNF placement is NP-Hard. Therefore, there is a need for placement methods that can cope with the complexity of the problem and find appropriate solutions in a reasonable duration. The primary purpose of this study is to provide a taxonomy of optimization techniques used to tackle the VNF placement problems. We classify the studied papers based on performance metrics, methods, algorithms, and environment. Virtualization is not limited to simply replacing physical machines with virtual machines or VNFs, but may also include micro-services, containers, and cloud-native systems. In this context, the second part of our article focuses on the placement of Containers Network Functions (CNFs) in edge/fog computing. Many issues have been considered as traffic congestion, resource utilization, energy consumption, performance degradation, etc. For each matter, various solutions are proposed through different surveys and research papers in which each one addresses the placement problem in a specific manner by suggesting single objective or multi-objective methods based on different types of algorithms such as heuristic, meta-heuristic, and machine learning algorithms.**

***Index Terms*—Virtual network function, container, placement, 5G network slicing, cloud native.**

I. Introduction

A. Motivation and New Trends

NOWADAYS, with the tremendous growth of mobile service demands, operators have to provide low latencies and high throughput, hence the need to rethink traditional physical architectures. Therefore, 5G tends to be virtualized, eventually, the 5G core network, i.e., the set of transmission and switching media in which the most crucial part of the traffic is processed, will no longer be carried by physical equipment as in 4G architectures, but it will be supported by the software. This virtualization is based on software-defined networking (SDN) and network function virtualization (NFV) technologies [1]. It offers many advantages and helps to provide personalized connectivity services through network slicing technology. Thus, the mobile operators will be able to agilely activate the functions required for each service, adapting the network sizing and topology according to customer needs and cloud properties: low or high throughput, low latency, high reliability, more or less distributed architecture, etc.

Mobile operators are currently experiencing a surge in 5G adoption, prompting service providers to implement the most recent Stand Alone (SA) 5G Core (5GC) [2]. Like any invention, many years of design and redesign were involved in making the 5G vision a reality, and the development process is still ongoing today. Consumers may now have access to 5G technology, but the backroom work and rework continue.

One of the issues that 5G has to deal with is figuring out how to effectively use infrastructure as a service to furnish flexibility, security, dependability, and, eventually, profitability. What began as an experiment in leveraging NFV to run VMs on hardware has swiftly evolved into VMs on shared computing and storage farms (Cloud). An orchestration layer was crucial to reduce operational overheads, but resource utilization with VMs remains a sticking point, and switching to a Container architecture (CNFs) became necessary. The architecture of the 5G mobile network control plane can be a hybrid architecture between cloud-native applications and virtualization [3]. As shown in Figure 1, the network virtualization approach transforms traditional network appliances with non-standard hardware into software-based virtual machines installed in standard equipment. Network functionalities that were previously developed as monolithic programs are now split down into smaller micro-services and delivered as containers in both public and private clouds using the cloud-native method [4]. These micro-services containers are orchestrated and supplied automatically using Continuous Integration and Deployment (CI/CD). Smaller micro-services are currently being provided by independent software suppliers who used to provide full-fledged network operations.

Manuscript received 27 August 2022; revised 9 January 2023; accepted 19 March 2023. Date of publication 31 March 2023; date of current version 12 December 2023. Mohsen Guizani acknowledges the financial support of Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI), Abu Dhabi, UAE under project number: 8481000021. The associate editor coordinating the review of this article and approving it for publication was R. Pasquini. *(Corresponding author: Essaid Sabir.)*

Wissal Attaoui is with the the Intelligent Network and Mobile Charging Department, INWI Corporate, Casablanca 20190, Morocco (e-mail: w.attaoui@ensem.ac.ma).

Essaid Sabir and Halima Elbiaze are with the Department of Computer Science, University of Quebec at Montreal, Montreal, QC H2L 2C4, Canada (e-mail: sabir.essaid@uqam.ca; elbiaze.halima@uqam.ca).

Mohsen Guizani is with the Machine Learning Department, Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE (e-mail: mguizani@ieee.org).

Digital Object Identifier 10.1109/TNSM.2023.3264005

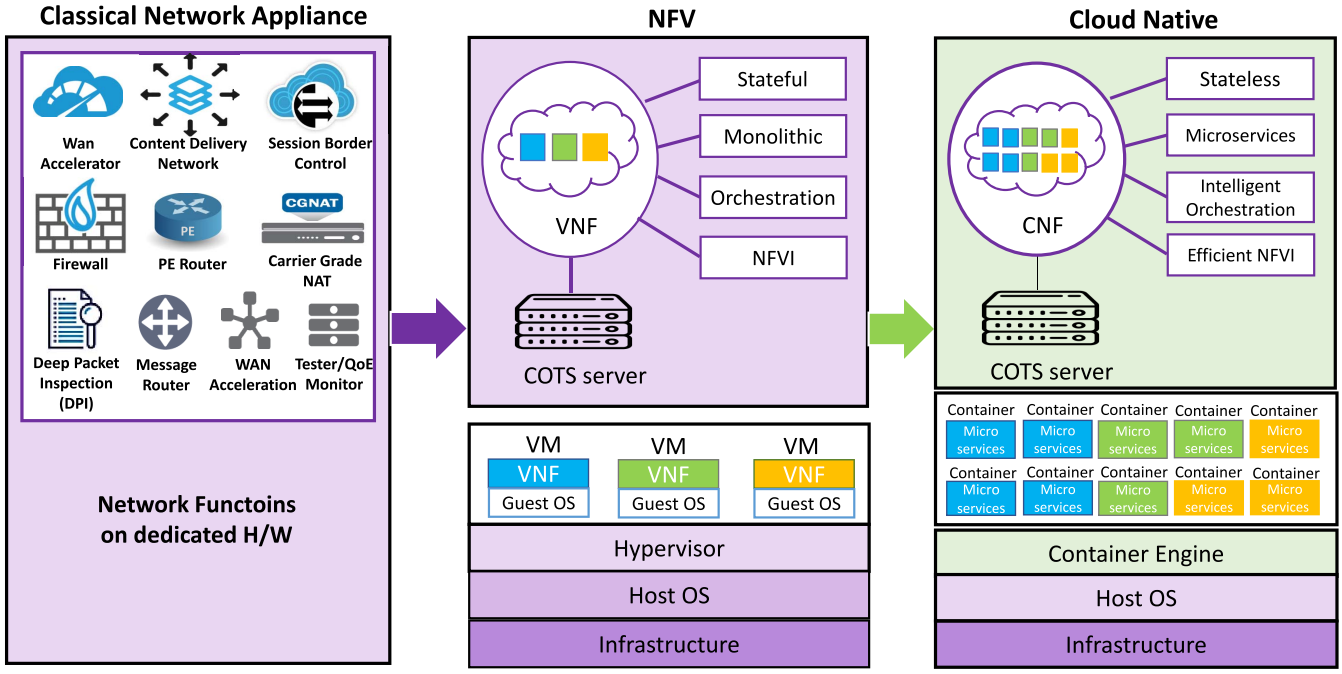

Fig. 1. Monolithic architecture, VNF virtualization architecture and CNF Cloud architecture.

The 5G deployment will be gradual (i.e., initially, 4G and 5G will coexist seamlessly, as was the case for 3G and 4G). In addition, pooling physical infrastructure through virtualization techniques opens the way to create a universal 5G network core that is agnostic to the type of access (i.e., wireless, wireline, etc.). Therefore, this will ensure homogeneous management of the operator's network. In this context, network slicing will enable mobile network operators to manage different virtual networks on the same physical network infrastructure [5]. The "slices" have features adapted in real-time to the users' needs (i.e., capacity, latency, reliability, etc.), whereby three new slicing services will be supported by 5G:

- *Massive Machine Type Communications (mMTC):* present the communications between many objects with varying quality of service (QoS) requirements to match the exponential increase of connected object's density;
- *Enhanced Mobile Broadband (eMBB):* is related to ultra-high-speed outdoor and indoor connection with uniform quality of service, even at the edge of the cell;
- *Ultra-reliable and Low Latency Communications (uRLLC):* ultra-reliable communications are used for critical needs with very low latency and increased responsiveness.

5G communication, with its enhanced characteristics such as high bandwidth and low latency, is ideally positioned to meet the expectations of smart cities. According to predictions, cities will house half of the world's population by 2050 [6], resulting in billions of Internet of Things (IoT) devices. There will be two issues in this situation. On the one hand, smart IoT services' real-time nature is seriously compromised. Massive numbers of smart IoT devices receiving data packets cause congestion in the central cloud which may degrade the QoS. On the other hand, inflexibility in computing resource allocation is highlighted. It is challenging to allocate computing resources to smart IoT devices as they have various characteristics.

Including virtualization in 5G will help to reduce latency, provide high speed, increase scalability and improve energy efficiency. However, the real problem relies on how to place a VM or Virtual Network Function (VNF) in a cloud infrastructure optimally.

Virtualization provides the flexibility to quickly move a VM from a specific host to another without turning it off. Therefore, it can provide dynamism on VM placement with a marginal performance impact [7]. A virtual instance can be added or deleted at any time. Despite its considerable advantages, this dynamic can lead to sub-optimal or volatile configurations of virtual networks. In previous research (before 2010), the cloud controllers manage the VM placement. However, current studies are increasingly focusing on the native dynamics of VM placement as each VM has its lifetime and can experience load changes during its life cycle. Therefore, it is necessary to define the VM placement requirements, know the status of each instance and correctly determine the essential constraints required to guarantee high performance. Operating system-level virtualization based on containers is a relatively modern technique of virtualization. Containers use the host operating system and do not require a separate one for each container resulting in lower hardware requirements than VMs. Orchestrators are required for managing and defining rules and constraints of container placement and performance [8].

B. On-Demand Computing in 5G

During the last decade, the cloud has evolved into a successful computing paradigm for delivering on-demand services over the Internet. The cloud data centers adopted virtualization technology to manage resources and services

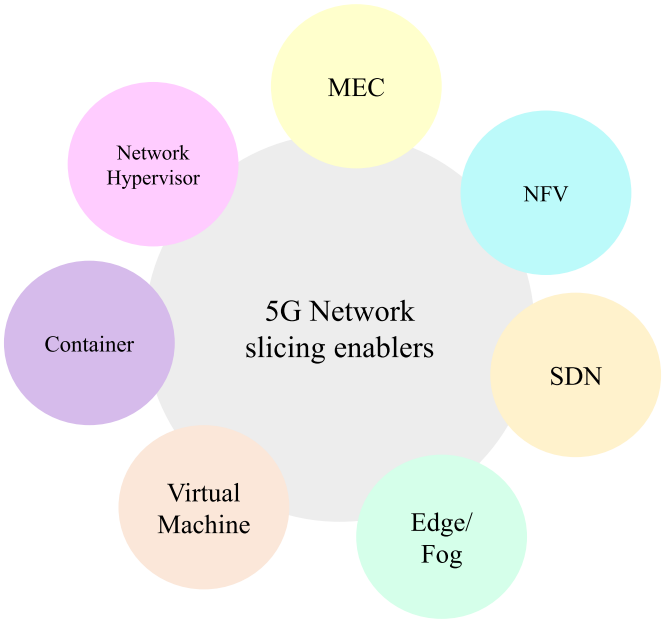

Fig. 2. 5G Network Slicing enablers.

efficiently. Advances in server virtualization contribute to the cost-efficient management of computing resources in cloud data centers. Cloud computing allows consumers to use on-demand computing resources in the form of instances (VMs or containers) rather than building physical infrastructure. These resources can be quickly delivered and handled effortlessly by cloud computing providers. It offers many interesting benefits for manufacturers and end-users, such as on-demand virtual resource provisioning, self-service capability, resource pooling, high elasticity, flexibility, and scalability. In addition, edge and fog computing are coined to complement the remote cloud to meet the service demands of a geographically distributed large number of IoT devices.

5G requires a complete makeover compared to previous generations due to exigent requirements and fast-changing new use-cases. The slicing-based one-network-fits-all strategy must meet these complex and ambitious goals. Network slicing (NS) enables service providers to construct and configure their networking infrastructure to meet their own needs and tailor it for various complex scenarios. Cloud computing is likely an inextricable aspect of 5G services, serving as a superior back-end for apps running on accessing devices. In this way, VMs and containers may execute VNFs in a chained configuration to offer a flexible 5G network service or application, laying the foundation for 5G network slicing. Figure 2 shows the 5G network slicing enablers, including SDN, NFV, Mobile Edge Computing (MEC), cloud/Fog computing, network hypervisors, VMs, and containers. Despite the numerous advantages and dynamic nature of VNF placement to create 5G network slices, if the placement is not chosen carefully, this may lead to sub-optimal or unreliable results.

Therefore, many issues should be considered in VNF, and CNF placement (e.g., power consumption, traffic, resource wastage, security, QoS, cost, etc.). For each matter, various solutions are proposed through different surveys and research papers in which each one addresses the placement problem in a specific manner by suggesting various methods based on different types of algorithms such as heuristic, meta-heuristic, deterministic, and machine learning algorithms.

C. Methodology

In this paper, we aim to find the optimal and most effective virtual resource (i.e., VNF, CNF) placement approach among all existing solutions by: i) determining the different problems that may arise in placement; ii) classifying the proposed solutions according to their adopted approach and objective functions; and iii) providing relevant insights that can help researchers to choose the most suitable solution regarding their game constraints.

This section provides an exhaustive description of our methodology, which is outlined in Figure 3. We provide a selection process of current research work to study a large part of the most relevant literature on virtual resource placement. We select the papers based on keywords, publisher, and abstract reading.

1) Keywords Search: The selection process of relevant articles started with a search of research articles from Google Scholar database [scholar.google.com] with at least one of the following selected keywords in the article title: VNF placement, CNF placement, Container placement, VM placement, resource allocation, VNF/Container migration, network slicing, placement in 5G cloud-native, network function splitting and placement, placement in fog computing, edge computing. These keywords search step results in 294 research articles.

2) Publisher: Considering the high number of results from the previous keyword search step, the literature selection process focused on research articles published in the following relevant publishers: ACM, IEEE, Elsevier, MDPI, Springer; Wiley and Taylor & Francis. The percentage of articles per publisher in the studied universe is summarized in Figure 4. This publisher filtering step results in a reduction from 294 to 217 research articles in the literature.

3) Abstract Reading: Considering the 217 resulting articles from the publisher filtering step, an abstract reading was performed in order to identify only the most relevant articles that specifically study the VMP problem. After the abstract reading, 176 research articles were selected from the literature. Finally, short papers (i.e., research articles with less than six pages) were removed from the selected literature, resulting in 162 selected articles of the virtual resource placement literature for the detailed the study presented in this survey.

4) Criteria for Inclusion/Exclusion: In order to limit our scope, we considered only works published in journals and conferences in the last 8 years (between 2016 and 2022). A selected paper must focus on VNF/CNF placement in different scenarios related to 5G network. It is important to highlight that we have removed works that consider basic VM placement in cloud computing as it was already treated in our previous papers [9], [10]. The inclusion criterion includes studies that comprise the methods, algorithms, frameworks and models for VNF/CNF placement issues.

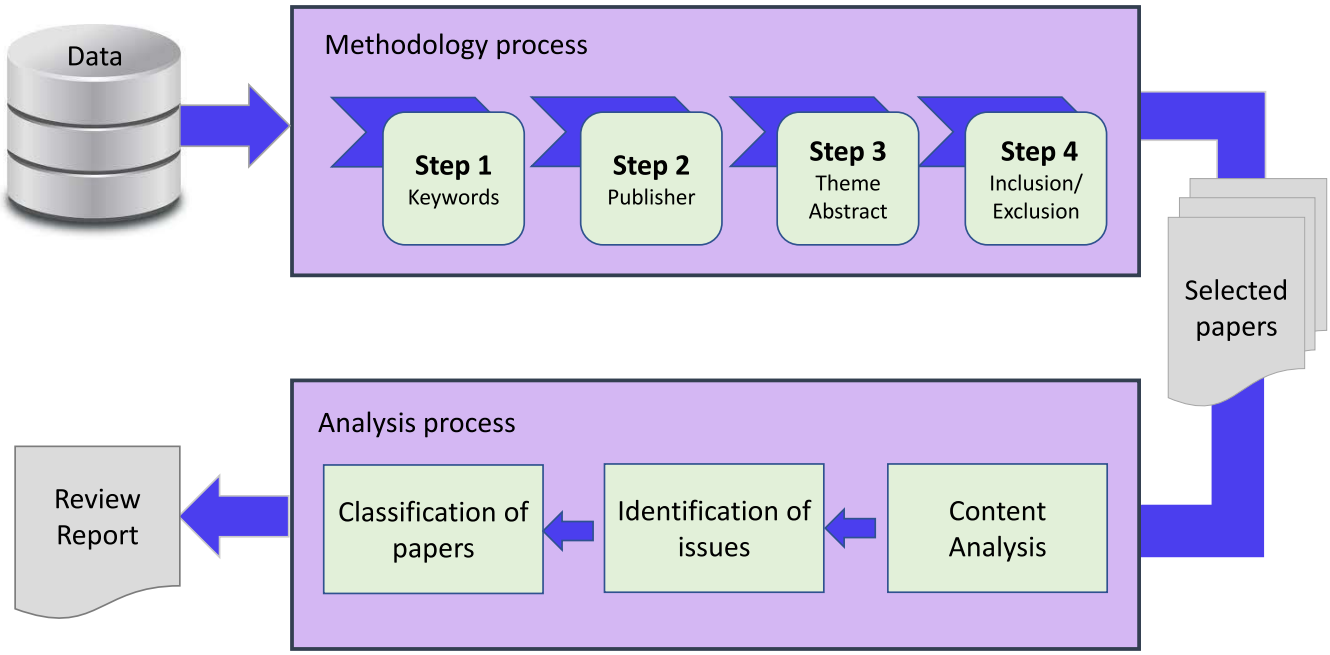

Fig. 3. Literature review process.

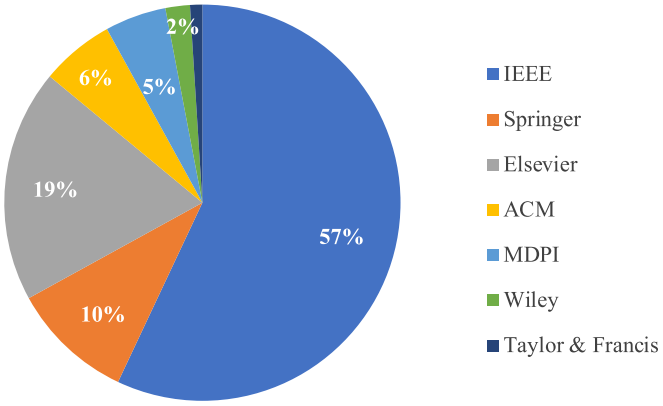

Fig. 4. Percentage of articles per publisher.

5) Research Questions: In this paper, the selected literature will answer the following questions:

- What is the specific issues related to VNF and Container placement in 5G Edge computing?
- What are the methods, algorithms and solutions proposed to solve these placement problems?
- What is the purpose of deploying containers instead of VMs or VNFs in 5G?
- From the studied papers, what are the best algorithms for VNF or Container placement?

D. Our Contribution

The scope of this study includes research results on efficient placement of computing resources (i.e., VNFs, CNFs) on various scenarios at different environment, providing constraints and metrics needed to improve the performance and QoS of hosted applications. This paper presents a detailed overview of VNF/CNF placement in evolving cloud infrastructures managed by mobile/fixed network operators and cloud providers. This article reviews the related literature over the period 2016-2021. The contributions of this survey are summarized as follows:

- We present an overview of 5G network slicing and VNF placement challenges raised in the literature, and we provide a summary of proposed solutions.
- We classify the proposed solutions according to the objective functions and the adopted techniques.
- We discuss the recent advances for 5G and the convergence toward containerized 5G architecture, where we highlight the CNF placement problems and solutions.

E. Related Work

For academic purposes, we summarize in Table I, some previous surveys that have already explored the field of VM, VNF and Container placement in cloud computing. Schardong et al. [11] present a literature review of VNF forwarding graph embedding (VNF-FGE) where the classification of VNF placement solutions is based on whether adopting online or offline approaches and the algorithms are categorized into exact, heuristic, and meta-heuristic. Alashaikh et al. [12] provide a detailed review of VM placement approaches by classifying the solutions based on adopted algorithms or models (i.e., heuristics, meta-heuristics, matching, and Multi-Criteria Decision Making (MCDM)). Li and Qian [13] discuss the network function placement and orchestration frameworks. Demirci and Sagiroglu [14] propose a taxonomy of VNF placement solutions where optimization methods are divided into four types: linear programming, non-linear programming, heuristic algorithms, and Machine Learning (ML) algorithms. However, they focus only on cost, energy, and latency minimization. Oleghe [15] highlights the concept of

TABLE I
A COMPARISON OF OUR WORK WITH EXISTING SURVEYS BASED ON KEY PARAMETERS

Ref	Type of virtual resource			Infrastructure			Classification schemes		
	VM	VNF	Container	Cloud	Edge	Fog	Objective functions	Heuristic and meta-heuristic algorithms	Machine learning based orchestration
[11]	✗	✓	✗	✓	✗	✗	✓	✓	✗
[12]	✓	✗	✗	✓	✗	✗	✓	✓	✗
[13]	✓	✓	✗	✓	✗	✗	✓	✗	✗
[14]	✗	✓	✗	✓	✗	✗	✓	✓	✓
[15]	✗	✗	✓	✓	✓	✗	✓	✓	✓
[16]	✓	✓	✗	✓	✓	✓	✓	✓	✗
[17]	✓	✓	✗	✓	✓	✓	✓	✓	✓
This survey	✗	✓	✓	✓	✓	✓	✓	✓	✓

container placement and migration in edge computing. He studied the scheduling problem from the provider perspective by listing the frameworks and algorithms used to model and solve the container placement issues. Buyya and Srirama [16] explore research proposals for network slice orchestration in various platforms, including VM/VNF placement in the cloud, fog, and edge computing. Sonkoly et al. [17] introduce a comprehensive survey of computational unit placement strategies in edge infrastructures; the studied papers were classified based on mathematical models, objective functions, and application structure. This survey has the same scope as ours, but the authors do not address the placement of CNFs in the framework of 5G. To the best of our knowledge, our paper is the first one handling the network function placement problems and highlighting the importance of containers; we also emphasize the importance of machine learning algorithms to solve the complex problems of virtual resource orchestration, such as multi-dimensional and dynamic workload characterization and auto-scaling.

From industry perspective, vendors can provide platform virtualization software and network functions (RAN/Core software, MEC platform) based on standardization (e.g., ITU-R/3GPP/ETSI/O-RAN). As far as open-source projects are concerned, there is a strong alignment with NFV architecture and RAN disaggregation. The Open Network Automation Platform (ONAP) [18], Open Source MANO (OSM) [19], Mosaic 5G [20] and Open5GCore [21] projects stand out in this scenario, led by the Linux Foundation, the ETSI and EURECOM. For example, the project under development by ONAP is natively oriented by the O-RAN initiative that supports dynamic network function placement. Nevertheless, the other open source projects still lack conscious concepts of network function placement either in RAN or 5G Core Network.

F. Article Structure

This paper is organized as follows. Section II presents the basic concepts related to cloud computing and virtualization. This paper is divided into two main parts. The first one, presented in Section III, addresses the problems of VNFs placement and presents a classification of the proposed solutions based on their objective functions and algorithms. Similarly, Section IV presents a new perspective towards cloud-native by handling the placement of containers in the cloud, edge, and fog computing. Section V delivers some concluding remarks and future directions. The organization of the paper is illustrated in Figure 5.

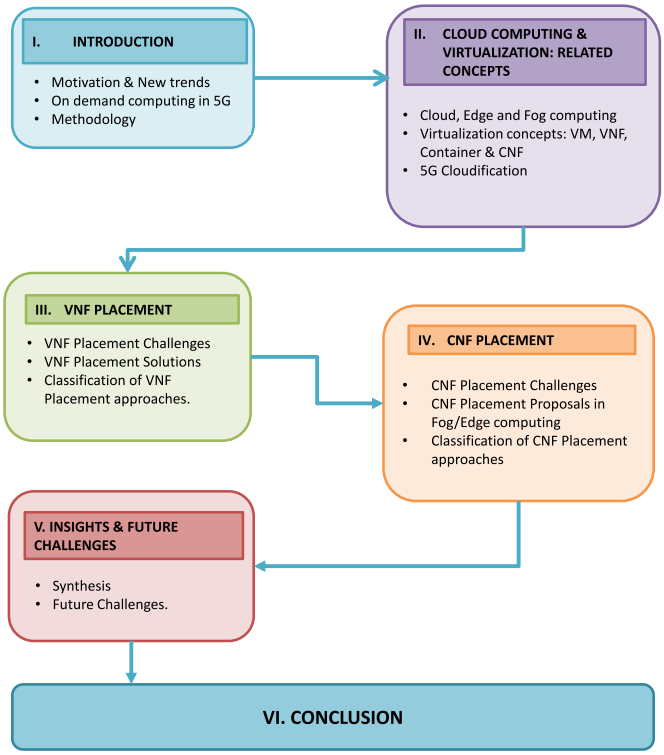

Fig. 5. Pictorial view of this paper.

II. RELATED CONCEPTS OF CLOUD, VIRTUALIZATION AND 5G

The main idea of IoT is that everything can be connected to the Internet at any time where a glut of objects (e.g., smart cameras, wearable devices, environmental sensors, home appliances, and vehicles) are connected and produce massive volumes of data. These data may be collected, integrated, processed, and analyzed to create smart cities, infrastructures, and services that improve people's quality of life. Existing IoT designs are highly centralized, relying primarily on moving data analytics, processing, and decision-making to cloud solutions. However, latency, network traffic management, computational processing, and power consumption can all be affected by data management and processing in the

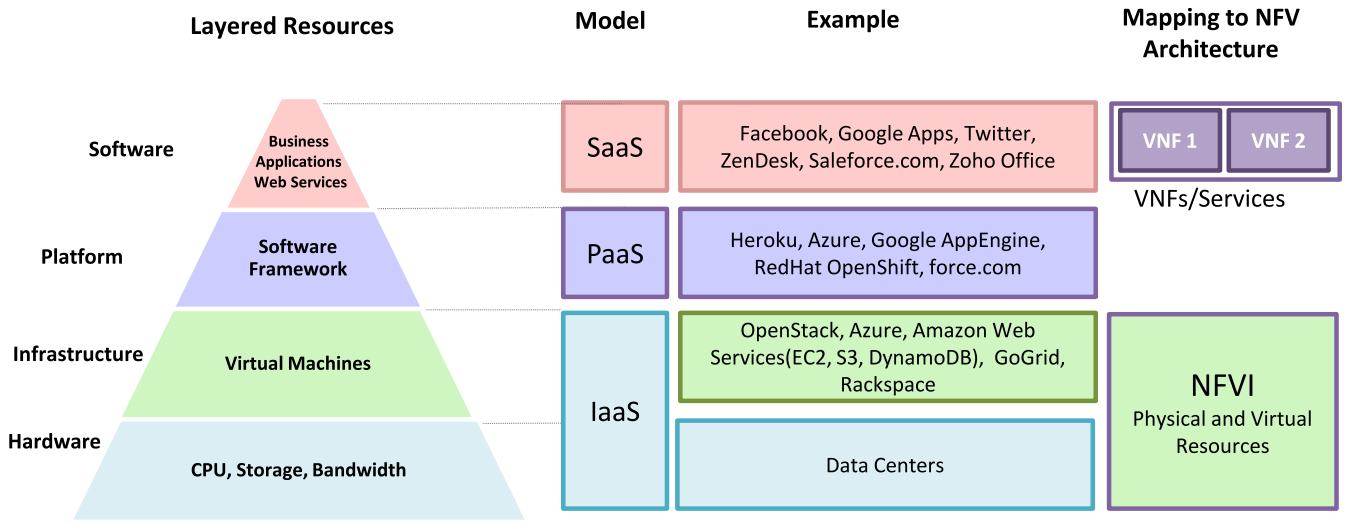

Fig. 6. Cloud computing service models and their mapping to part of the NFV reference architecture.

cloud. Moreover, in many applications that need low latency, such as health monitoring and emergency response services, the delay created by transmitting data to the cloud and subsequently back to the application can significantly influence the system's performance. Data fusion, data trends, and various decision-making approaches allow data processing closer to where data is generated and help to minimize the quantity of data transferred to the cloud, reducing network traffic, bandwidth, and energy consumption. In addition, smart city applications such as smart health, security, and traffic control will benefit from a more agile response that is closer to real-time. Therefore, this section contrasts the cloud computing paradigm with the more sophisticated paradigms used to bring compute, storage, and control capabilities closer to where data is generated in the IoT (i.e., fog and edge computing). Also, it provides a summary of crucial virtualization concepts (i.e., VM, Container, VNF, CNF).

A. Cloud, Edge and Fog Computing

1) Cloud Computing: Nowadays, the term cloud computing is already widespread. With the pandemic of covid-19 and the growth of remote working, companies have been forced to look for this type of solution to stay competitive.

IT industries have defined cloud computing from different business perspectives but the most commonly accepted definition among experts is the one provided by the National Institute for Standards and Technology (NIST), which considers cloud computing as *"a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction"* [22]. According to IBM, *"Cloud computing is on-demand access, via the Internet, to computing resources, applications, servers (physical servers and virtual servers), data storage, development tools, networking capabilities, and more hosted at a remote data center managed by a cloud services provider"* [23], cloud computing converts the IT infrastructure into a utility that provides a dynamic and scalable service-oriented IT architecture. Put simply, cloud computing includes both applications provided as services over the Internet and the data center hardware and software that deliver those services. Cloud computing is a global concept of anything that requires the provision of hosted services over the Internet. Infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) are the three primary types of these services [24] as shown in Figure 6. According to NIST [25], there are five enabling characteristics of cloud computing: on-demand self-service, elasticity, resource pooling, measured service, and broad network access.

Today, cloud computing is encountering increasing challenges in satisfying the stringent requirements of new IoT applications. Latency and network bandwidth are two major issues. Future IoT solutions based on AI and emerging technologies rely significantly on the cloud as it provides nearly infinite storage and computing capacity [26]. These talents are required to turn the massive volumes of data created by the IoT into intelligent knowledge and directives. However, traditional cloud computing models are reaching their limits and are unable to handle this massive amount of data. Two new paradigms have been proposed to address these weaknesses, namely fog computing and edge computing, allowing additional computational resources (such as storage, networking, and processing) to be brought closer to the network's edge.

2) Fog Computing: CISCO introduced the concept of fog computing in 2012 to expand cloud capabilities closer to the network's edge *"Fog computing is a highly virtualized platform that provides compute, storage and networking services between end devices and traditional cloud computing data centers, typically, but not exclusively located at the edge of the network"* [27]. Since then, other definitions have emerged under various circumstances and setting. The fog is a layer that stands between the edge and the cloud, bringing the cloud closer to the IoT data processing nodes resulting in a cloud-to-things continuum that reduces latency and network bottlenecks while maintaining data privacy.

In [28], fog computing is considered as *"a paradigm to complement the cloud for decentralizing the concentration of*

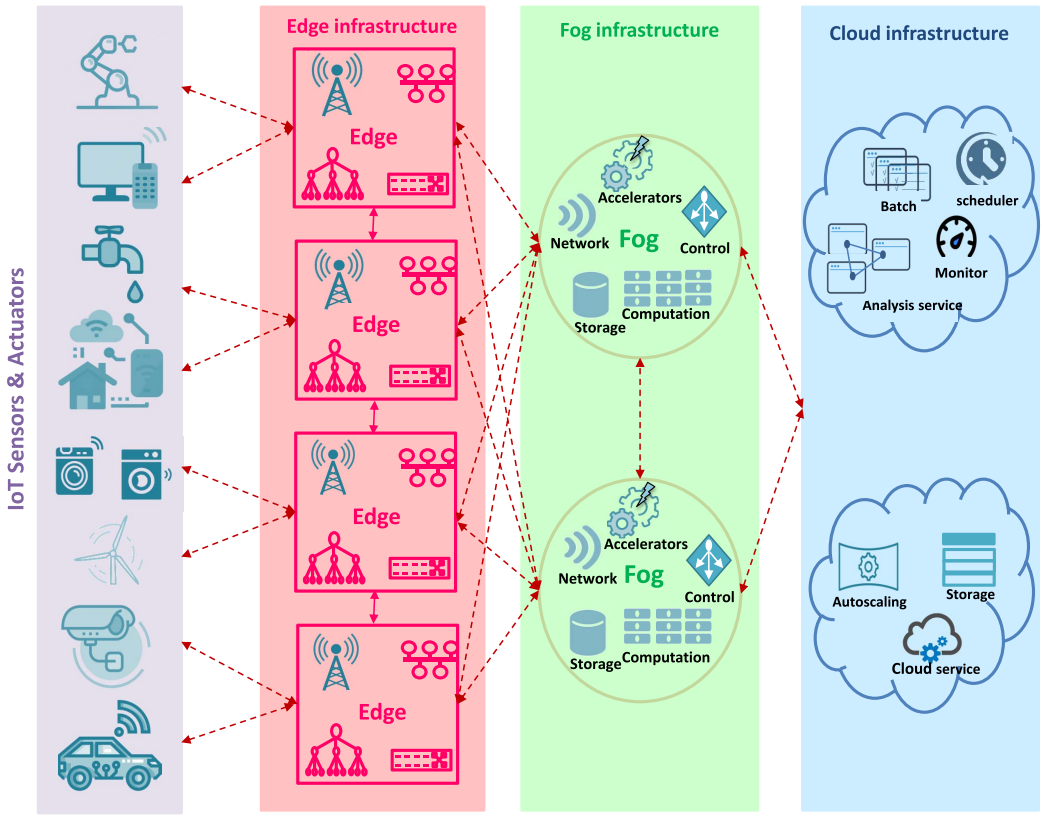

Fig. 7. Cloud, Fog and Edge computing.

computing resources (for example, servers, storage, applications, and services) in data centers toward consumers in order to improve service quality and user experience."

According to NIST, *"Fog computing is a layered model for enabling ubiquitous access to a shared continuum of scalable computing resources. The model facilitates the deployment of distributed, latency-aware applications and services, and consists of fog nodes (physical or virtual), residing between smart end-devices and centralized (cloud) services."* [29].

Fog computing differs from the conventional computing models by the following features: (1) geographical dispersal, (2) contextual location awareness and low latency, (3) heterogeneity, (4) interoperability and federation, (5) real-time interactions, and (6) federated fog cluster scalability and agility [30]. In addition to these six fundamental qualities, fog computing is frequently related to (7) Wireless access predominance and (8) mobility support.

3) Edge Computing: Some literature considers edge computing as a synonym of fog computing [31], [32], but there are some substantial differences. In [33], edge computing refers to *"enabling technologies allowing computation to be performed at the edge of the network, on downstream data on behalf of cloud services and upstream data on behalf of IoT services"*. Edge computing concentrates on the things aspect, while fog computing concentrates more on the infrastructure aspect. The objective of edge computing is to move specific computing resources from the cloud to heterogeneous devices at the network's edge [34]. According to CISCO [35], edge computing refers to bringing computational resources closer to data-generating devices, whereas fog computing refers to the physical implementation and management of this architecture at the cloud's edge.

Fog computing uses a multi-layered architecture to supply hardware and software operations, allowing dynamic re-configurations for diverse applications while performing intelligent activities. Edge computing provides a direct delivery service by running specific applications at a fixed logical location [36]. Edge computing tends to be restricted to a small number of peripheral devices (e.g., BS, home gateways, edge routers), whereas fog computing is hierarchical.

The fog and edge computing architectures enable the computing and storage capabilities of the network infrastructure to be leveraged for the deployment of IoT services, thereby making these services closer to the end-users. However, the network devices are heterogeneous with low computational capacity, covering a wide geographical area, and have to address the mobility of IoT users. In this way, the problem of virtual resource placement becomes more complex in terms of optimizing various parameters such as minimizing energy consumption, enhancing IoT QoS, reducing traffic congestion, and decreasing cost. MEC provides cloud computing capabilities to content providers.

B. Virtualization Concepts: VM, VNF, Container, CNF and SFC

The way network services are delivered to end-users has been transformed by NFV. Individual network services are

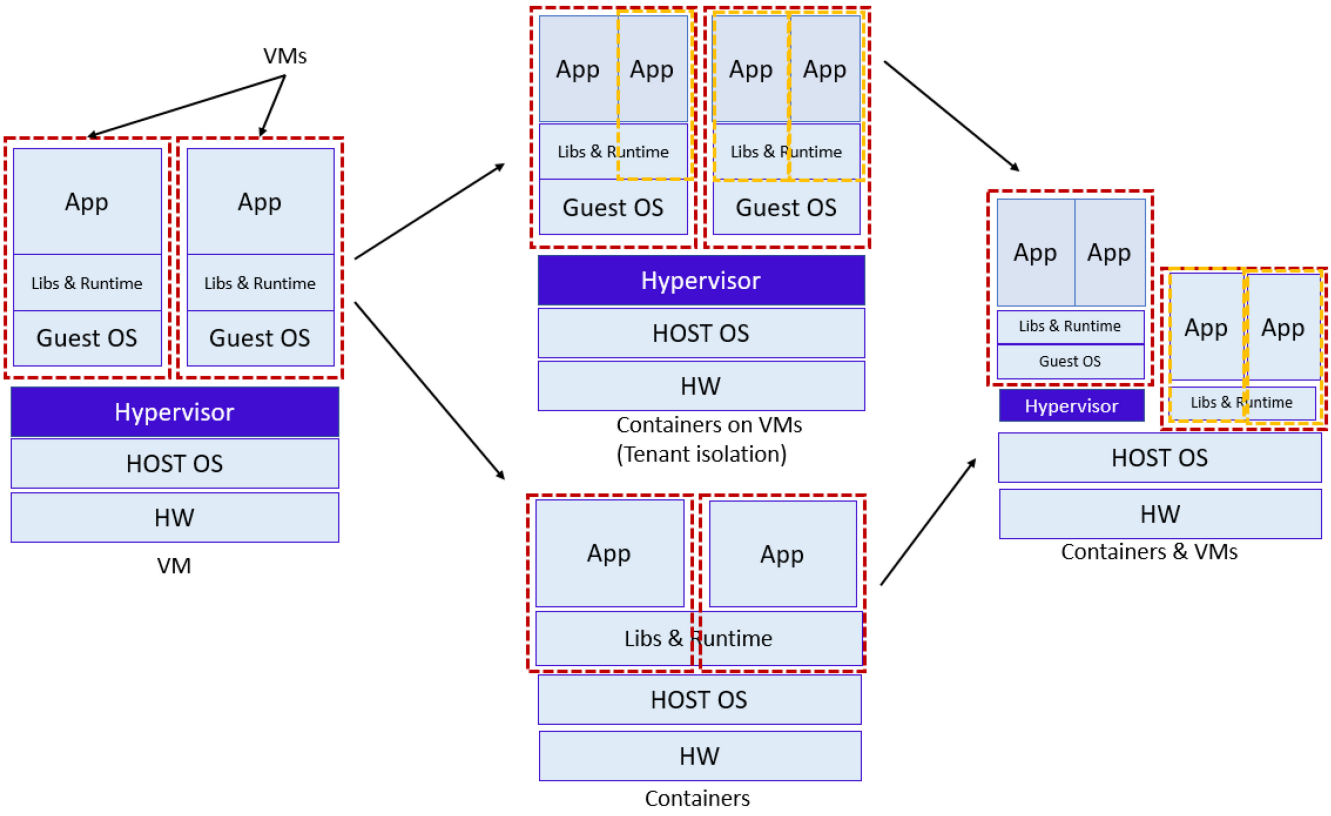

Fig. 8. VM and Container deployment.

now provided as software-based virtualized entities known as VNFs, which are dissociated from costly and specialized middle-boxes [37]. It is a piece of software that handles network tasks, including routing, switching, firewalling, and load balancing. It also eliminates the need for separate proprietary and specialized hardware from vendors, allowing network services to be executed on generic or Commercial-Off-The-Shelf (COTS) hardware with varying degrees of computing, storage, memory, and network interfaces. VNFs can be hosted using two virtualization technologies, VMs, and containers.

Although VNFs are part of a conventional network design, they still have constraints as digital telecom providers progress toward offering more flexible services. When switching from physical components to VNFs, providers merely uninstalled the embedded software systems from the devices and established a large virtual machine. However, without efficiently optimizing and placing these virtual resources, this can create inefficient single-use appliances and even impact the quality of service [38].

Furthermore, the weight of VMs may restrict VNF efficiency for large-scale 5G or edge deployments that require agility, scalability, and minimal overhead. Therefore, telecom operators tend towards adopting a cloud-native approach using distributed and centralized locations that help to ensure efficiency, scalability, and reliability.

The key component of the cloud-native approach is the usage of containers rather than VMs. Containers enable users to bundle software (for example, apps, functions, or micro-services) with all the files required to operate it while sharing access to the operating system and other server resources. This method allows the enclosed component to be moved across environments (development, test, production, etc.) and even between clouds while maintaining performance. Table II shows a comparison between VMs and containers.

As summarized in Table II, containers are lightweight alternatives to VM-based hypervisors [39] and are characterized by OS-level virtualization. In containers, a physical server is virtualized to enable autonomous applications and services to be deployed on a remote server. Unlike their VM-based counterparts, containers do not require hardware indirection and run more efficiently on the host operating system, allowing for greater application density.

In recent years, hyper-scale clouds have evolved customer expectations around infrastructure consumption. The market has shifted to containers, micro-services and on-demand infrastructure powered by APIs and automation.

- Containers: From a basic perspective, system-level virtualization permits multiple virtual instances of an operating system to run simultaneously on a single server on top of the hypervisor. On the other hand, containers are isolated and share OS kernels among all containers (see Figure 8). Containers are widely used to optimize hardware resources, run multiple applications, and improve flexibility and productivity. Thus, a container is considered as an operating system-level virtualization technology that can be deployed on VMs or PMs and primarily used to provide a secure and isolated environment.

TABLE II
COMPARISON BETWEEN VM AND CONTAINER

Features	Virtual Machine	Container
Performance	Suffer from a low overhead as the instructions from the Guest to the Host OS are translated	Provide close-to-native performance compared to the Host OS.
Startup time	It takes several minutes for VMs to boot up.	Can boot containers in a few milli-seconds.
Storage	VMs require even more space as a whole OS kernel as it has to install and run the related programs.	Containers take lower space, as the basic operating system is shared.
Isolation	Hardware isolation	Operating system isolation
Kernel	Each VM run its own OS	Containers share the same kernel
Benefits	Fully isolated	Lightweight, Native performance, less memory requirement, more portable
Drawbacks	Heavyweight, limited performance, Large memory requirement, less portable	Higher fault domain

- Micro-services: A micro-service is an architectural and organizational pattern where every application function has its own service. These services are deployed in containers, and these containers speak with each other via APIs. The use of micro-services allows IT systems to be organized in the form of instances that can be added/removed on-demand in order to increase/decrease the scalability of their functions. However, companies that run thousands of micro-services in containers on the cloud didn't have a simple way of managing them, whereby the need for orchestration and management.
- Orchestration and automation: A few popular orchestration solutions are built to monitor the system, trigger the container's status, and balance the load between the active application instances, etc. The orchestrator used for CaaS has a direct influence on the available functions to cloud service users. Nowadays, the container virtualization market is dominated by three orchestration tools: (i) Docker Swarm multi-source cluster management and orchestration tool marketed by Docker as a native tool for managing docker clusters and container operations; (ii) Kubernetes [40], an open-source project from Google that provides a centralized system for scaling, managing containers, and automating deployment; (iii) DC/OS (the Distributed Cloud Operating System) [41], an open-source distributed operating system that enables the management of several machines in the cloud from a single interface; It allows the deployment of containers, distributed services and legacy applications in these machines and also ensures networking, service discovery, and resource management to help running and communicating services with each other.

Cloud-native networking functions (CNFs) are an extension of VNFs that are intended and constructed to run in containers [42]. This containerization of network architectural components enables a range of services to run on the same cluster and provides easier integration of already deconstructed applications while dynamically routing network traffic to the appropriate pods. CNFs can address some primary constraints of VNFs by shifting many of these functions into containers. Containerizing network components allows administrators to control how and where functions are executed across clusters.

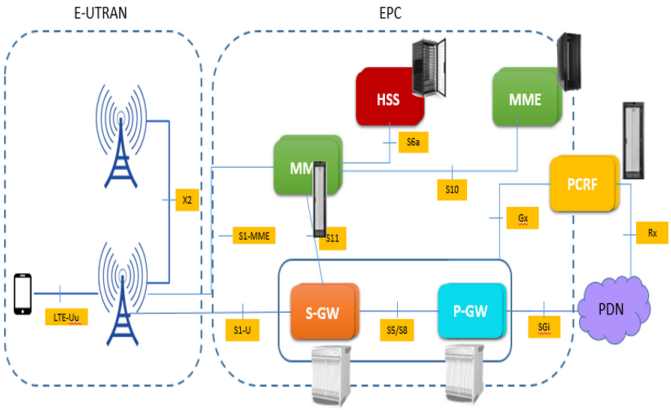

Fig. 9. Legacy Core architecture.

Another trending paradigm in telecom networks that deserves to be highlighted is Service function chaining (SFC). SFC is a mechanism that permits different service functions to be connected to each to construct a service allowing carriers to benefit from the virtualized software-defined infrastructure. SFC is used to configure VNFs/CNFs into one logic chain with specific requirements (i.e., throughput, latency, and error rate) to deliver good QoS/QoE in a 5G network. The SFC concept composes and imposes the order in which service functions are invoked for a particular service. SFC is crucial for the granular management of virtual networks and will involve the use of VNF forwarding graphs (VNF-FGs). This will be highly required due to the growing number of deployed VNFs and QoS-sensitive services, as well as the maintenance needed for the inter-VNF point-to-point connection.

C. 5G Cloudification

In the traditional architecture of mobile networks (i.e., 2G, 3G, 4G), some physical functions are used to provide voice and data services to customers. The network is composed of a radio access network, which connects the customer to the antennas, a backhaul part, and an IP backbone (IPBB) which consists of high speed switches and routers for connecting the user to the core network. The customer data processing feature is provided by a chain of several Physical Network Functions (PNFs) as seen in Figure 9, such as Home Subscriber Server (HSS), Mobility Management Entity (MME), Policy Charging Rule Functions (PCRF), Packet Data Network Gateway (PGW), Serving Gateway (SGW), etc.

With the high demand for data traffic in very dense areas where customers require low End-to-End delay, small latency, and high reliability, the physically oriented architecture of the legacy network is unable to meet all these ambitious

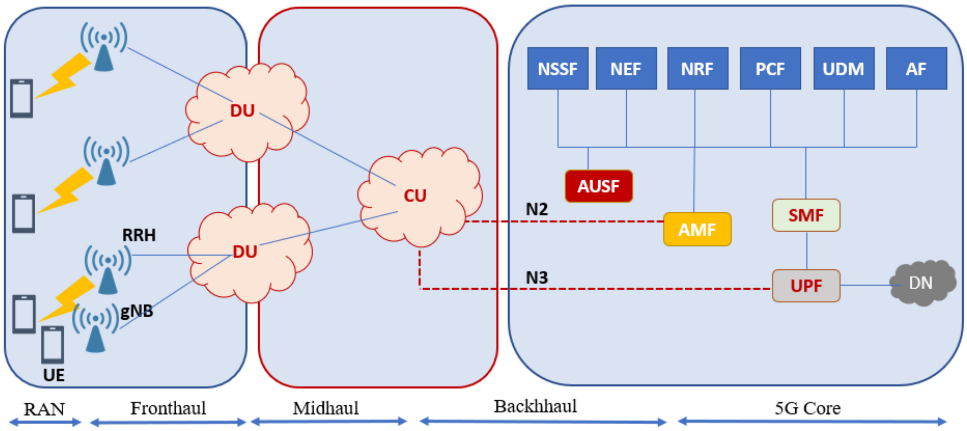

Fig. 10. End-to-End 5G mobile network Architecture.

targets. Therefore, the future generations of mobile wireless communication systems (i.e., 5G, 6G) is expected to meet these goals. 5G will provide more reliability and flexibility through 5G core network (5GC) and New Radio (NR) [43]. A cornerstone of 5G is network function virtualization and containerization, which is the basis for telecom operator data centers and network orchestration.

Communication service providers struggle to cope with the inherent growth in traffic, improve their customers' experience, and develop solutions to offset significant CAPEX and OPEX challenges. They adopt NFV technologies and migrate from physical hardware platforms to virtualized ones based on softwarization and cloudification. With NFV, network functions are virtualized and are named VNFs. They are deployed as VMs (virtual machines) with hypervisors such as Linux KVM or VMware vSphere on Commercially Off-The-Shelf hardware (COTS), allowing operators to avoid vendor dependency.

For a lightweight implementation of functions, regarding cloud-native architecture, mobile operators prefer to use Containerized Network Functions (CNFs) instead of VNFs as CNFs are more lightweight and elastic than VNFs. CNFs can operate in a micro-services architecture that provides a dynamic, flexible, and scalable architecture for 5G. A Cloud-based architecture should be employed in a manner to handles VNF/CNF functions efficiently while considering placement issues.

Each service includes a chain of services formed by several VNFs/CNFs connected to each other. VNFs/CNFs can be placed in different core data centers. The most promising movement is driven by the industry's Open RAN (ORAN) initiative, which focuses on open and interactive solutions. It implements an open interface between three network elements Central Unit (CU), Distributed Unit (DU), and Radio Unit (RU), using hardware and software defined functions. The ORAN brings cloud agility to the radio access part, which increases scalability, reliability, and availability.

To cope with the traffic increase, mobile phone operators have to include diverse small cells by adding eNodeBs composed of indoor and outdoor types of equipment (i.e., remote radio heads (RRH) and baseband units (BBUs)) and linked via mobile Fronthaul/Backhaul to provide a 5G infrastructure called 5G crosshaul (see Figure 10). This new design [44] enables placing VNFs/CNFs, provisioning the required network and computing resources in a flexible, cost-effective, and abstract manner.

ORAN provides the ability to place network functions at different locations along the signal flow. This option is called "functional split". Because of the different throughput and latency requirements, radio access function splitting policies will affect the sizing of the backhauling network, and thus the placement of core network functions and the configuration of edge computing application servers. In the following, we will handle the VNF/CNF placement in radio access and core networks.

VNFs/CNFs, in 5G network architecture (see Figure 11), provide complete core network functions as Home Subscriber Server (HSS), Mobility Management Entity (MME), Access and Mobility Management Function (AMF), Session Management Function (SMF), and Policy Control Function (PCF), etc. The 5G SMF is an immediate component of the 5G service-based architecture (SBA). The SMF is mainly responsible for interacting with the decoupled data plane, creating, updating, and deleting Protocol Data Unit (PDU) sessions, and handling the session context with the User Plane Function (UPF). Both the UE and the gNB use the Next Generation Application Protocol (NGAP) to carry Non Access Stratum (NAS) messages on the N1 or N2 interfaces to initiate a new session request. The 5G SMF is an immediate component of the 5G service-based architecture (SBA). The SMF is mainly responsible for interacting with the decoupled data plane, creating, updating, and deleting Protocol Data Unit (PDU) sessions, and handling the session context with the User Plane Function (UPF) through the N4 interface by using the Packet Forwarding Control Protocol (PFCP). The AMF gets these requests and processes anything related to connection or mobility management, while sending the session management requests to SMF on the Nsmf interface. It determines which SMF is most suitable to manage the connection request by polling the Network Repository Function (NRF).

During session initiation or update, the SMF send control requests to PCF through Npcf interface, along with the

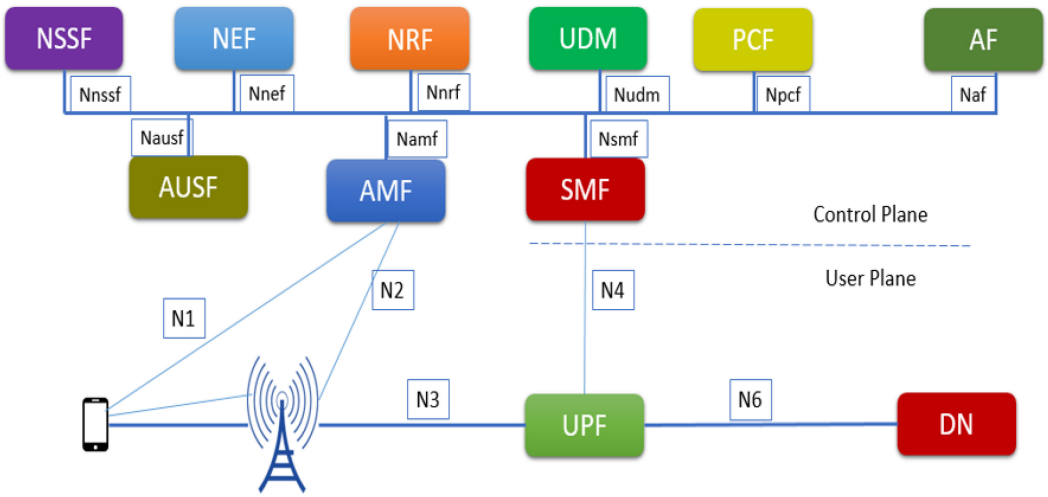

Fig. 11. 5G Core Architecture.

subscriber profile informations stored in the Unified Data Management (UDM) (Nudm). All service base application elements are connected to the Service Based Interface (SBI) message bus, for example the PCF provides the policy charging rules.

III. VNF Placement

Nowadays, the telecom world is evolving exponentially to become purely virtualized. Therefore, virtualization is considered as the key element of 5G using SDN, NFV, and MEC technologies to build virtual partitioning of mobile radio access, virtual core network, and network slicing.

A VNF is an implementation of a network function that can be a firewall, a router, a load balancer, or even a mobile core network component. Regarding VM placement problems already discussed in previous papers [9], [10], [45], the open question is how telecom providers will work around the issues of placing VNFs in the 5G network, and the challenging task relies on where, when, and how to place the VNFs.

In this context, network slicing offers several significant advantages, which are valuable for the design of next-generation networks [46]. Slicing provides an agile VNF placement, improving network performance and decreasing operating costs. It involves deploying multiple logical networks as separate business transactions on a shared physical infrastructure [47]. Various architectures have already been proposed to provide evolved 5G infrastructures, [47], [48], [49] which offer the capabilities to support the required diversity of services, scalable deployments and network partitioning.

In [48], a 5G-ready architecture model and an NFV-based network slicing are presented to provide scalable VNFs and deliver 5G slices that meet customer requirements. In the same way, authors in [49] offer a new architecture for open cloud-based 5G Communication that treats the network slicing as a brain wave in the cloud-based Radio Access Network (RAN), aiming to increase the scalability of current RAN systems.

In network slicing, each slice has its own envelope that is a compromise related to the target usage, and its characteristics should be tailored to the chosen environment. For example, in a single 5G system, the network slicing technology can provide connectivity for smart counters using a network slice that connects IoT devices to a data service with high availability and reliability, with a given latency, throughput, and security level. At the same time, it can provide another network slice with very high throughput and low latency for an augmented reality service.

Therefore, 5G has a flexible structure where network slices assign capacity, velocity, and coverage resources separately. In this way, network slicing allows the coexistence of multiple vertical services over the same physical infrastructure. Based on [48], and [49], the network slices architecture is divided into three layers as illustrated in Figure 12:

The business layer is a market of applications and network functions used to provide various scenarios with different features (e.g., high mobility, speed, IoT). It creates a slice that encrypts all information required from the service layer to provide the requested function.

The service layer manages, configures, and scales the operational set of services according to their particular use case qualifications defined in the “slice manifest”.

The infrastructure layer manages the re-configurable green cloud system in real-time and applies virtualization for high-speed services. The slicing in 5G typically drives two new insights, i.e., the service layer and the network slice orchestration, to supervise the life cycle of slices.

The slice orchestration is a complicated matter that can be divided into intra-slice and inter-slice problems. One of the intra-slice orchestration’s crucial characteristics is the efficient placement of VNFs, including initial placement (static) and online placement (dynamic) throughout slice run-time. An intelligent placement may reduce latency and operating costs and increase resource utilization and network performance.

Network slicing is a collection of interconnected VNFs and physical functions over a common multi-domain infrastructure to support a specific service. Its performance depends directly on the efficient placement of VNFs. For example, slices lacking low latency have to be placed close to end-users. Therefore,

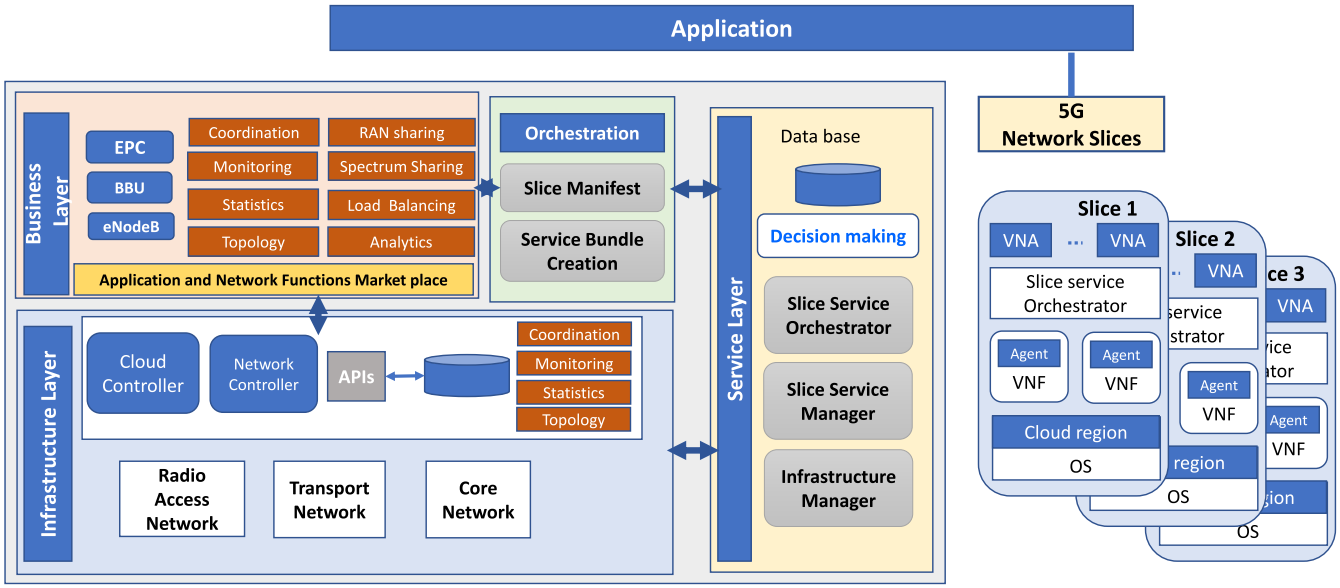

Fig. 12. 5G Network Slicing Architecture ([48], [49]).

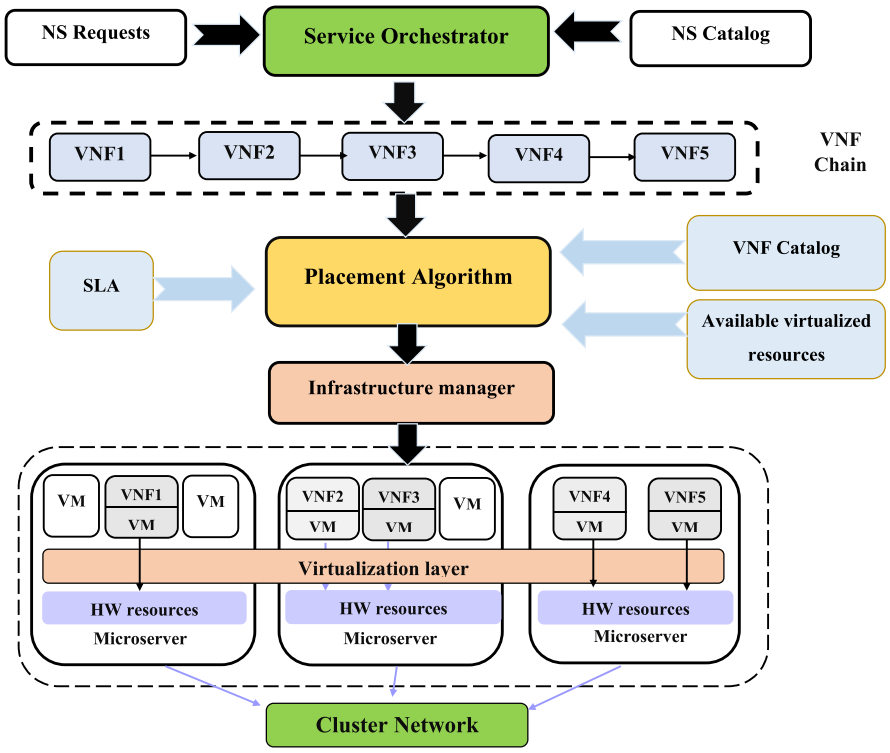

Fig. 13. VNF Placement process.

considering the end-to-end performance of a specific network service, VNFs must be placed in the best locations. We present in Figure 13, a scenario of VNF placement in a 5G distributed edge cloud where NS is defined as a collection of VNFs required to deploy a complete 5G mobile service.

Network slices can cross various network domains, including access, core, and transport. In this context, extensive efforts have been performed to address the problems of functional placement. This section describes the VNF placement issues and challenges in terms of energy, power consumption, capacity, latency, and security. Considering the advantage of the NFV's ability to place VNFs anywhere and anytime easily, several VNF placement strategies are proposed for different NFV orchestration settings. We also present a classification of VNF placement solutions in 5G based on their objective functions.

A. VNF Placement Challenges

The VNF placement requires multi-objective functions such as reducing cost, minimizing the end-to-end latency, reducing

energy consumption, ensuring reliability, etc. However, the trade-offs between these objectives can lead to several conflicting issues, as placing several VNFs in the same device can cause scalability problems. For example, reducing the number of active hosts can increase network link aggregation, which affects network latency. In addition, minimizing the network latency can be impacted by VNF redundancy deployment where a cost-efficient solution is required [50]. Besides, the network power consumption must be minimized while meeting latency requirements [51]. Moreover, combining resource allocation and traffic routing raises a significant issue for VNF placement [52], where the decision to place VNFs has a critical influence on the efficient use of resources and the energy consumption in DCs. Two questions need to be addressed: how each host's computing capacity should be shared between the VNFs? and which physical machines should run the required VNFs?

VMs or containers can allow VNFs to be placed on low-cost devices to perform services at the network edge. When VNFs are placed near to end-users, this can minimize End-to-End (E2E) latency, response time, and even unneeded core network usage. The VNFs must be placed appropriately to handle end-user movements and address the traffic dynamics resulting in highly variable latency on network links. In addition, VNF placement in 5G networks faces substantial reliability and latency difficulties, resulting in customer dissatisfaction and revenue loss. The VNF deployed as a VM co-located with many other VMs on the same server might impact network performance when dealing large traffic loads. Inequitable network resource sharing and VM traffic load might therefore cause increased latency.

The security risk in VNF placement is another issue that must be considered, as malware attacks can lead to substantial financial damage and loss of customers [53].

According to the literature, the different issues related to the VNF placement are energy consumption, cost, resource use, traffic, and security in order to solve the overall problem of end-to-end performance and latency (see Figure 14). VNF placement includes network functions placement, VNF forwarding graph, and VNFchains placement.

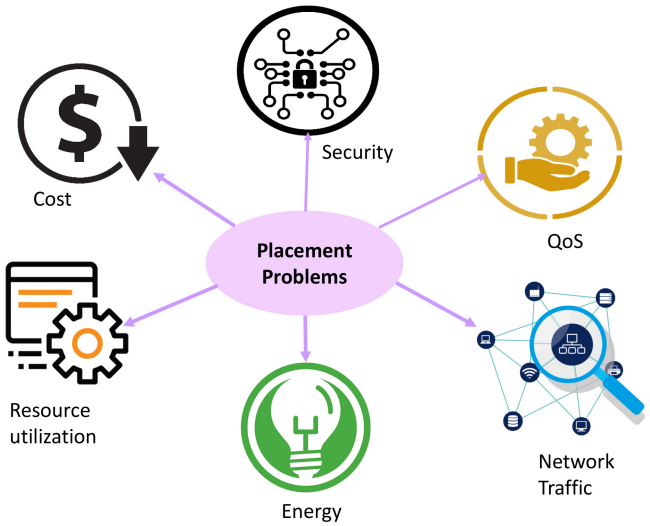

Fig. 14. VNF Placement issues.

B. VNF Placement Solutions

Mobile operators can provide specific services (e.g., social networking, video streaming, augmented reality, etc.) by chaining VNFs and routing traffic among them. This section introduces an overall classification of the different VNF placement approaches proposed in the literature. The majority of related works handled the VNF placement problem as a multi-objective trade-off with latency. VNF placement can be carried out on different network domains: access, core, and transport. Extensive efforts have been made to solve the functional placement problem.

An optimization objective is used to measure specific aspects of the solution generated by an algorithm. In some papers, the optimization objective may consist of a single objective, while others may be multi-objective depending on the optimization needed for the problem at hand. Generally, the more objectives used in the cost function for the considered optimization problem, the more complex the decision-making process becomes. Therefore, different trade-offs are normally set in place to balance the performance of the proposed algorithm and the quality of the generated solution. The following are considered the most common objectives utilized in the cost function definition of container placement problems.

1) Energy Consumption: Energy consumption is a significant concern for data centers. With the growth of network traffic, the power consumption of the infrastructure also induces a high cost for NFV providers. Therefore, from cost control and environmental protection point of view, reducing power consumption is very crucial.

In [54], the authors attempt to find the optimal placement of service function chains, considering the optimization objectives for different network slices and the functional split between the central cloud and the distributed radio access point. They propose an optimization framework for placing RAN services based on an Integer Quadratically Constrained Programming (MIQCP) model and Maximum Satisfiability (MaxSAT). The problem is considered as a multi-objective approach to reduce the network latency, minimize the number of utilized nodes, reduce the power links capacities, and maximize the data throughput on the network links by keeping the bandwidth for future demands exclusively to support eMBB services. The authors analyze some scenarios using uRLLC and eMBB slices with various resource specifications. Experimental results prove that the MIQCP model is faster than MaxSAT in finding optimal solutions; however, this latter is more suitable for highly constrained problems.

Reference [55] addresses the joint problem of VNF placement and CPU allocation decisions in the 5G network. Decisions are taken sequentially; first, the authors propose a heuristic algorithm called MaxZ that provides the deployment decisions. Then, they make CPU allocation decisions by solving the convex optimization problem of minimizing the maximum ratio latency based on the fixed results of the MaxZ heuristic. Regarding performance evaluation results, MaxZ outperforms greedy and affinity-based algorithms.

MEC in the NFV environment is considered as the 5G uRLLC enabler regarding its capacity to minimize energy consumption and end-to-end latency. The uRLLC service comprises several VNFs, where VNF placement is similar to VMP since VNFs are virtual instances performing network functions. In [56], the VNF placement accommodated for uRLLC services is formulated as an optimization approach aiming to minimize latency and maximize service availability. This model is solved using a genetic meta-heuristic algorithm. Experiment results show that this heuristic algorithm gives solutions near-optimal in less time than an exact algorithm provided by CPLEX.

In the same way, authors in [51] propose an energy-aware placement solution based on a Robust Optimization (RO) approach to minimize energy consumption while satisfying latency constraints. They use constraint modeling with Soyster heuristic model [57] to solve the problem. Their purpose consists of placing each VNF in the available network slice to the convenient VM of the service chain in a joint cloud radio architecture. The overall energy consumption is defined as the sum of energy in all assigned VNFs, and the end-to-end latency is defined as the total delay of NS, VNF, the processing delay, the path delay of NS between VNFs, and the link delay.

In [58], authors introduce a new concept of "accessible scope," defined as the group of servers used to serve a request. Instead of searching the whole servers to find the optimal placement, all servers can be divided into groups where each group can serve one specific request. The primary purpose of [58] is to minimize the server energy consumption induced in VNF placement while improving time efficiency taking into account resource constraints. Authors use the accessible scope to narrow the searching space of VNF placement and therefore reducing the searching time. They execute the Multi-Stage Graph method with the accessible scope constraint (MSGAS) to see how the size of the accessible scope affects the acceptance ratio, energy consumption, and bandwidth usage. Furthermore, the results of the algorithms with and without the accessible scope requirement demonstrate that the ones with the accessible scope constraint reduce the runtime significantly, especially for large-scale networks.

NFV can offer flexible placement of VNFs in the underlying data centers. However, the VNFs placed on the same server may experience performance interference due to shared memory and computing resources. In [59], authors propose an approach that considers energy consumption and performance interference in VNF placement. The problem is formulated as a bin-packing problem that is NP-complete. For a homogeneous environment, a First-Fit (FF) heuristic algorithm is proposed to solve the complex problem with a lower bound. For heterogeneous cases, an efficient solution named Deep Deterministic Automatic Placement (DDAP) based on Deep Reinforcement Learning (DRL) is proposed to achieve better placement. Simulation results prove that DDAP outperforms existing approaches such as FF and Ant Colony System (ACS) in terms of reducing energy consumption and running time.

In the same way, for dynamic SFC placement, two policy-based Reinforcement Learning (RL) algorithms, Proximal Policy Optimisation (PPO2) and Advantage Actor-Critic (A2C), are proposed to minimize the energy consumption while considering the availability levels required by the customer and SLA [60]. The model is formulated as a Markov decision process where SFC requests are processed sequentially. The RL algorithms yield better results than the greedy algorithm in terms of energy consumption and acceptance rate.

For the same purpose, an RL technique is used to design a VNF placement policy in an NFV architecture aiming to handle the VNF forward graph embedding problem (i.e., the resource placement in the underlying network) [61] while reducing the overall power consumption. This paper is considered an extension of Neural Combinatorial Optimization (NCO) by including constraints (e.g., SLA, latency, bandwidth, and resource utilization) in the problem definition. Simulation results prove that combining AI models with heuristic algorithms can improve the heuristic itself without requiring expertise and knowledge.

2) Cost: Authors in [62] handle the VNF placement problem in service chains to ensure a reduction in operation and traffic costs. They propose an algorithm called SAMA that merges a "sample-based Markov approximation" with matching theory seeking an effective way to reduce operational and network traffic costs. This strategy first selects the nodes where VNFs can be deployed then places the VNFs in a way to minimize the total cost. Results prove the performance of SAMA in terms of reducing the cost by up to 19% compared to non-coordinated solutions.

The high cost of network power is also a significant challenge for VNF placement. In this context, authors in [63] propose a new joint placement approach for VNFs and their associated chains over the cloud computing environment. Their system performs joint node and link mapping using the extended eigen-decomposition of the request and infrastructure graphs. The main objective relies on maximum matching with minimum bipartite graph (BG) cost. This suggested algorithm achieves better performance than the greedy algorithm because it is fast stable, and its execution time depends only on NFV infrastructure size.

According to the literature, several works have addressed cost reduction by properly utilizing computing resources in cloud-based mobile core networks, seeking the optimal placement of VNFs in the same data center. For initial VNF placement, FZ. Yousaf et al. [64] propose two algorithms called Vertical Serial Deployment (VSD) and Horizontal Serial Deployment (HSD), aiming to minimize the overall cost. For highly overload profiles, the HSD can efficiently reduce the average throughput per active server since the load is shared equally across all racks with the increasing number of active servers. However, VSD leads to unbalanced distribution; servers in specific racks can be 100% utilized while other racks are underutilized or not used. To address this problem, a new automated NFV orchestrator based on machine learning [65] named zero-touch orchestration (z-TORCH) is proposed to improve the quality of management and orchestration systems by providing an optimal placement of VNFs with minimal monitoring cost.

Authors in [66] propose a dynamic placement of VNFs based on an online efficient scaling algorithm to minimize the

network cost. They consider a network composed of multiple time slots with prediction stages. In each prediction stage, they apply a forecasting approach based on Fourier-Series to decide whether new demands exist in the new time slot. This online learning mechanism, based on Upper Bound Confidence (UCB), aims to reduce costs by withdrawing frequent changes in the network topology.

In [67], authors propose a heuristic approach based on bin-packing for minimizing cost in VNF placement while considering coverage, mobility, battery consumption, reliability, and low latency constraints for deploying services over a volatile 5G network. The proposed heuristic outperforms a state-of-art mobility-aware algorithm, achieving near-optimal deployments in terms of cost while enhancing convergence speed to the solution (thus increasing the number of time-feasible solutions) and reducing the number of requested handovers.

Authors in [68] handle the problem of joint traffic routing and VNF placement for a multi-cast service request to reduce both the VNF and link provisioning costs. The optimization model is formulated as a Mixed Integer Linear Programming (MILP) problem. Therefore, heuristic solutions are proposed for single path and multiple-path routing scenarios to minimize the embedding cost and provide a flexible placement and routing while ensuring low latency.

Telecom providers should handle real-time requests in the small end-to-end latency to satisfy user demands with good QoS in the 5G network; Therefore, MEC has been deployed to minimize the customer experienced delays. In [69], an SDN/NFV-enabled MEC architecture is proposed to reduce the deployment cost. However, the incurred cost for VNF placement and resource allocation (VNFPRA) in MEC nodes must be considered. The VNFPRA is formulated as a MILP problem and solved using a genetic-based heuristic algorithm to minimize the global resource cost, including the allocation cost, the computation cost, and the link usage cost. Simulation results confirm the efficiency of the suggested Genetic Algorithm (GA) based VNFPRA compared to FF and RF placement algorithms.

Similarly, in [70], authors propose a new approach for VNF placement issue for SFC using replica in the software-defined cloud, named VNF and Replica Placement (VNFRP). This approach reduces the overall SFC placement cost, service response time, energy consumption, and link bandwidth utilization. First, the problem is formulated as an ILP. Then the VNFRP heuristic algorithm is used to find the optimal placement by dynamically placing the VNFs of the SFC in the same or different nodes based on the SFC placement cost and the minimum link bandwidth.

Several approaches have been proposed to address the complexity of adjusting and placing VNFs in physical networks regarding the high number of nodes and links in DCs. The most of existing solutions focus on static placement that it initiated only if a change happened. For example, when an event occurs or some areas are busy at a certain time, it will create an overload on some servers, and therefore a VNF placement/readjustment procedure is implemented, which may cause latencies and affect the QoS. Consequently, the dynamic placement of VNFs should be deeply investigated due to the ever-changing resource availability in cloud DCs and the continuous mobility of users. The majority of VNF placement/readjustment solutions focus on optimizing objectives such as power consumption resource utilization, but they ignore important features as latency and service level objective (SLO) penalty violation cost. In this context, authors in [71] propose a Machine learning approach named MAPLE that divides the substrate network into a set of separate clusters, to reduce the complexity of VNF placement and adjustment. For the network partitioning problem, they apply the k-medoids clustering technique and a statistical technique to optimize the selection of the initial group of medoids. This helps to enhance the quality of the clusters while reducing clustering time. The VNF placement/readjustment problem is formulated as an ILP that aims to simultaneously reduce the latency, the SLO violation cost, the resource utilization, and the cost of VNF readjustment. This dynamic model helps to provide administrators with real-time placement and readjustment decisions. For large-scale DCs, they design data-driven cluster-based placement and readjustment algorithms based on machine learning that intelligently remove some cost functions from the ILP optimization problem. Simulation results prove the performance of the proposed approach in terms of reducing latency and SLO violation cost compared to existing approaches as k-means, migration without clustering, and original k-medoids.

For stateful VNFs, it is challenging to find the optimal DC for placing active and standby VNFs while reducing their overall cost, including the cost of continuous state transfer from active to standby instances, as this may result in high bandwidth consumption or even network congestion. In this respect, a RL approach is proposed for placing stateful VNFs based on a joint reservation of active and standby resources while reducing the total placement cost [72]. Simulation results prove the performance of RL based VNF placement approach in terms of improving the acceptance ratio and reducing the overall cost compared to benchmark solutions (e.g., Node-Rank [73]) in online and offline scenarios.

3) Resource Utilization: The 5G network functions are placed on VMs that can be switched between different PMs. Power consumption can be reduced by stopping unused resources. However, it is not clear what are the required resources for the network function and whether placing more VNFs in a smaller number of physical resources can degrade the service user experience and violate service level agreements.

The fast and reliable resource allocation for network slices remains challenging since each slice requires specific functionalities such as bandwidth and processing power. VNF placement and CPU allocation decisions are influenced by routing decisions from one network node to another [55]. From this point of view, remarkable effort has been dedicated for combining VNF placement, resource allocation and routing problems [68], [74], [75], [76].

The implementation of an effective framework for resource allocation to network slices remains a very relevant issue. Hence, instead of worrying about how to place VNFs

individually and interconnect them, the cloud-native architecture efficiently allocates resources for network slices in terms of network bandwidth and cloud processing power [77].

In [78], the VNF placement is considered as a multi-objective optimization problem aiming to minimize the bandwidth dissipation and reduce the maximum link application simultaneously. Therefore, four genetic algorithms have been proposed using the frameworks of two existing algorithms, multiple objective genetic algorithms (MOGA) and non-dominated sorting genetic algorithm (NSGA-II): Greedy MOGA, Greedy NSGA-II, Random MOGA, and Random NSGA-II. Simulation results prove that Greedy-NSGA-II outperforms other algorithms.

In [79], the problem of VNF placement and chaining (VNF-PC) is handled by a flexible resource allocation approach aiming to minimize resource consumption in a small end-to-end delay. Authors propose a Mixed Integer Quadratically Constrained Program (MIQCP) named Flexible Resources Allocation Model (FRAM), which considers the tradeoff between resource allocation and latency, answering the question of how many resources should be allocated within the VNFs to meet the required latency. Results prove that FRAM outperforms the Strict Resource Allocation Model (SRAM) that does not consider the resource delay dependency.

In [80], a new VNF placement strategy is proposed for assigning adequate VNFs to hosts based on the total number of resources. First, before VNF placement, a periodic updating search method is applied to find the convenient host. Next, an on-demand fast VNF assignment upon request is used for placement instead of computing each time resource information.

In [81], authors propose a MILP approach to handle the joint VNF placement, resource allocation, and user association. Their optimization problem aims to reduce the service provisioning cost, minimize the effect of migration on customer's QoE, balance the resource allocation and optimize the transport network usage while guaranteeing data service requirements (e.g., latency, speed, etc.) in mobile edge computing.

The VNF Placement and Chaining Problem (VNF-PC) is one of the most challenging problems in NFV. It focuses on network resource allocation to provide end-user services such as massive IoT (mIoT). Nevertheless, these services must be supplied by an infrastructure that becomes progressively complex and heterogeneous with the growing number of network components and the exponential increase of the computational processing and runtime [82]. Therefore, as the VNF-PC is NP-hard to solve, authors in [83] first propose an ILP algorithm; however, this latter still suffers from high runtime. Second, they develop a hybrid optimization approach combining ILP with ML to minimize the number of network components and reduce the runtime in the Substrate Network (SN) through clustering strategies. The ML identifies first the patterns between requests and then decides which SN component will be used in processing. Two distinct clustering approaches are proposed: (i) based on the SN components' spatial location; and (ii) based on the SN components' historical resource usage. As a result, this hybrid strategy helps to minimize the runtime by up to 75% compared to exact methods and reduces the E2E latency without degrading the acceptance rate and provider's profit.

The challenges of network slicing and VNF placement are well debated in the literature but without considering the close relationship between the two concepts. In this context, the VNF placement over network slicing has gained attention by researchers addressing different objectives and constraints. As network slices may be implemented as a chain of VNFs, the two concepts are inextricably linked and must be explored together. The subject of resource allocation (capacity, compute, and storage) across slices has attracted much interest [84], [85], [86]. Some papers consider only spectrum resources (e.g., [85]), while others take into account also the computing resources required for VNF placement [84], [86].

In [87], the authors analyze the best VNF deployment and computational resource allocation in a hybrid two-clouds C-RAN architecture, taking into account various 5G service demands and distinctive 5G RAN functionalities. By setting limits on VNFs, the objective is to reduce the overall amount of computing resources. The problem is formulated as an ILP and solved using a standard solver. To cope with the computational cost of optimizing a large number of clouds and VNF chains, they present a simple low-complexity heuristic named Best Fit with IteRative Split Trial (B-FIRST) that tries to discover a suitable VNF placement solution with a small number of functional slices.

The NFV challenge in a C-RAN architecture is also addressed in [88], which examines six distinct criteria while formulating a C-RAN system that delivers VNFs on an edge data center. VNFs are put in the edge data center, and diverse network slices with varied requirements/constraints are considered (i.e., E2E service latency, E2E service reliability, E2E power consumption, computation capacity constraint, throughput constraint, and service admission probability). For multi-service 5G networks, authors in [89] present a new network function allocation approach that allows network functions to be deployed in a distributed computing environment based on service demands. The suggested technique includes both RAN and Core Network (CN) functions. Unlike existing systems, it provides an option capable of skewing the VNF placement based on service requirements, allowing for quick and straightforward operator-side network function deployment.

Despite the advantages of 5G networking technologies, there is a need for an automated and self-scaling orchestration system that is capable of placing VNFs dynamically to fully use MEC DCs for uRLLC services. In [90], a Deep Deterministic Policy Gradient (DDPG) RL algorithm is proposed to solve the dynamic placement of VNFs between edge and cloud network DCs. The proposed algorithm can provide the best VNF placement with respect to SLA requirement, E2E latency, and network resources compared to alternative solutions. They have proved the sustainability of DDPG for automated spatial resource allocation by migrating VNFs between cloud DCs and MEC DCs.

For large-scale networks, even the most advanced learning algorithms are unable to satisfy the complexity of VNF-FG placement. For example, the DDPG [91] is unsuitable for solving the high dimensional action space as a VNF-FG scheduling problem due to the restrictions of substrate network resources. In this regard, authors in [92] propose new approaches to find feasible solutions for the VNF-FG embedding problem by adopting the DRL technique to minimize the resource allocation while ensuring the QoS requirements. First, they propose a lightweight algorithm named Heuristic Fitting Algorithm (HFA) to deal with the efficiency of DDPG in large-scale space. The HFA determines an appropriate allocation policy of VNFs based on the proto action value received from the output of the DRL agent. Next, they propose an enhanced exploration DDPG named E^2D^2PG that provides new modules in addition to HFA, i.e., evaluator and enhanced exploration, for assessing the quality of the solution and enhancing the exploration of DRL agent. Simulation results prove the performance of E^2D^2PG compared to conventional DDPG and ILP. Similarly, in [93], authors suggest a DRL approach for multi-domain VNF-FG embedding. However, the results are only realized on tiny network architecture.

In [94], authors address the SFC allocation problem and provide a reinforcement learning approach for placing VNFs on an appropriate node that enhances VNF performance based on the physical network's load status. This algorithm provides good results compared to OpenDayLight (ODL) scheduler. However, this approach takes a long time to converge in a large exploration space.

In [95], an enhanced RL-based approach merged with an expert knowledge mechanism is proposed to circumvent a lengthy training procedure for VNF-FG embedding. This RL technique is based on Enhanced Q-Learning (EQL), aiming to accelerate the learning time, achieve load balancing, and improve long-term reward and performance. The EQL controls and learns the network based on the usage patterns of PMs. Simulation results, handled in large-scale networks, prove the effectiveness of EQL in terms of scalability, acceptance ratio, QoS, and acceptance gain, compared to ILP algorithms.

4) Network Traffic: Reference [74] targets the importance of combining VNF placement and path selection to maximize the served traffic demands and minimize network utilization. This problem is formulated with a mathematical program that systematically estimates a proper path length and reuses factors for each request. A usage-guided chain deployment algorithm is proposed to find a solution for optimal VNF placement in terms of reuse factor and proper path length. Simulation results prove that the proposed algorithm yields good results overcoming greedy-based and shortest-path-based heuristics. Therefore, the link capacity and resource availability should be jointly allocated in VNF placement.

In [75], the VNF placement is formulated as a mixed-integer linear programming problem. The main objective is to find an efficient placement of network functions with traffic routing among them while minimizing the CPU resource usage and the flows delay.

Reference [76] tackles VNF placement and chaining by proposing a new analytical approach based on the Eigen-decomposition method. This approach jointly manages VNF placement and traffic distribution where VNFs are placed, and traffic is spread over them all at once as tenant requests are processed collectively in the form of VNF forwarding graphs (VNF-FG). Simulation results prove that eigen-decomposition based heuristic is fast, stable, and serves more requests than the greedy heuristic algorithm.

Reference [96] also considers the VNF placement in 5G network slicing as an optimization problem aiming to achieve maximum throughput of accepted requests. A new heuristic algorithm named Adaptive Interference-Aware (AIA) is proposed to place VNFs automatically. The experimental results demonstrate the effectiveness of the proposed scheme in terms of enhancing the total throughput of slicing services such as video streaming and autonomous driving in comparison to other heuristic approaches. The AIA can also handle the traffic variations induced by VNF interference.

In [97], the authors propose a dynamic solution for joint VNF placement, traffic routing, CPU assignment, and VM activation to provide different vertical services in 5G network, considering the end-to-end delay as the primary KPI. To make this joint decision, the problem is formulated as MILP based on requests' arrival and departure times over the system's lifetime. Authors propose MaxSR, an efficient meta-heuristic method for solving the aforementioned problem for large-scale network situations based on near-future knowledge.

SDN and NFV technologies have been introduced as crucial paradigms for reaching the tactile Internet's low latency requirements in multi-access edge computing (MEC) cloud systems. In [98], the authors proposed a new approach for handling distributed SFCs toward low-latency tactile Internet applications, called Chain-based low Latency VNF ImplemeNtation (CALVIN). CALVIN aims to place VNFs in a distributed way with one VNF per VM. It applies fast packet input/output (IO) to prevent the metadata and batch processing of the classic Linux network stack.

Authors in [99] propose a solution for optimal VNF placement to provide eMBB services in NS by using spatial metrics of network topology. They suggest an architecture for 5G multi-tenancy networks with different software components to achieve smart decisions for VNF placement. Results prove the proposed prototype's performance in terms of computing the spatial measurements for a 5G multi-tenant network with 65538 mobile users in a small delay.

NS placement is known as an NP-hard optimization problem [100] that involves deciding which servers can host the VNFs forming the network slice and which pathways can be followed to direct traffic between these VNFs. Deep Reinforcement Learning (DRL) was recently employed in some network slice placement publications to solve this problem in a scalable fashion [101]. However, the majority of DRL research assumes a stationary environment, i.e., a static network demand. Traffic conditions in real networks are generally non-stationary and are vulnerable to large variations, such as traffic peaks, caused by unexpected events. This makes it

harder for the DRL algorithm to learn the context in which slices should be placed properly. In fact, the ever-changing network environment and policies may not be in harmony with the algorithms that require previous knowledge to develop the best solutions. In [102], a new solution adapted for non-stationary traffic conditions is proposed based on hybrid DRL-heuristic algorithms to deal with the traffic change. This framework combines Advantage Actor Critic and a Graph Convolutional Network (GCN). Four algorithms have been considered pure DRL, enhanced DRL (eDRL), Heuristically Assisted DRL (HA-DRL), and HA-eDRL. Results prove that in a real non-stationary network scenario, the suggested hybrid DRL heuristic technique is more reliable than pure-DRL.

5) QoS and Throughput: MEC and NFV have emerged as potential technologies for delivering low-latency IoT services in smart cities. IoT devices require computing services to meet the needs of everyday applications in smart cities. Each IoT service may be implemented as a service chain made up of several interconnected VNFs running on virtual machines. When considering VNF placement, QoS is a big deal since the MEC needs to deliver high-quality IoT services. Various IoT services require different levels of QoS. For example, monitoring-related services desire a cloudlet attaining better availability as the high-availability cloudlet provides steady computing help, while game-related IoT services would like a low-latency cloudlet. As a result, an exact QoS method is required to pick appropriate cloudlets for various IoT applications. However, QoS is impacted by multiple attributes such as availability, E2E latency, resource utilization, traffic congestion, the bandwidth of communication links, etc. Thus, it is challenging to assess QoS by multiple attributes. Furthermore, the network dynamically adapts to the state of cloudlets and switches in real-time.

Determining the QoS of each cloudlet in a dynamic network becomes even more difficult. While MEC and NFV can solve the problems of resource utilization and network congestion, they also bring new challenges. To address these difficulties, authors in [103] propose a multi-attribute-based QoS approach for VNF placement and service chaining in smart cities. The optimization problem aims to maximize the throughput subject to multiple constraints such as computing resource capacity, the bandwidth of communication links, and the QoS requirement of each demand. The multi-attribute problem is formulated as an ILP, and a heuristic algorithm based on the randomized rounding technique (RRH) is proposed to solve this problem. The authors also propose an algorithm named UFPH based on an unsplittable flow approach to handle VNF placement and service chaining challenges while meeting the extra QoS requirement of each type of IoT service. Simulations results prove that the two proposed algorithms (i.e., RRH and UFPH) outperform Greedy and Random algorithms in terms of high throughput and average QoS.

6) Security: Most previous works address different VNF placement problems without considering the resiliency of service chain embedding with the slicing concept. However, few works deal with it (e.g., [104], [105]). In [104], Xu et al. handle the cross-domain security problem in service function chain placement to reduce the end-to-end latency while satisfying resource constraints. This optimization problem is formulated using two ILP models for the inter and intra domains. Therefore, a heuristic algorithm is proposed to provide satisfactory solutions. In the same context, authors in [105] formulate the resiliency problem as an optimization problem aiming to reduce the maximum number of impacted service chains during a PM failure while meeting the slice-specific requirements and respecting the VNF placement constraints in the co-located network slices. Similarly, in [106], the authors provide three ILP algorithms to tackle the VNF placement issue while ensuring resiliency against single link, single node, and single-node/link failures.

C. Classification of VNF Placement Approaches

In the majority of real-world scenarios, VNF placement must be treated as an online problem. On the other hand, offline methods are very important to address issues that may not be obvious in online cases, where many requests are processed in sequential order. The orchestration and placement algorithms fully understand the requirements that will be executed simultaneously in an offline manner. VNF placement algorithms are categorized into online (dynamic) and offline (static) approaches. The VNF placement problems have inspired many researchers to develop several optimization methods. These problems have been identified as NP-hard. Based on the set of papers studied in this survey, we classify the search algorithms adopted to cope with the different VNF placement problems into three types: Heuristic, Meta-heuristic (random-based) search techniques, and Machine learning algorithms, which can be described as:

- *Heuristic:* Heuristics are problem-dependent models that are designed to solve a problem according to its specification. Despite the fact that these algorithms do not always ensure convergence to an optimum solution, they are capable of obtaining competitive solutions very quickly. Table III shows a classification of heuristic algorithms based on their key parameters and objective functions.
- *Meta-heuristic:* Meta-heuristics, considered as an extension to heuristic techniques, are a high-level problem-independent algorithmic framework that can identify near-optimal solutions by iteratively optimizing solutions based on a particular performance metric. Table IV provides the meta-heuristic algorithms adopted for VNF placement issues. Meta-heuristics allow to efficiently solve the VNF placement problem. They can incorporate new objectives or constraints very easily without changing the solution, unlike heuristics. It is also important to notice that some research works demonstrate the convergence of these heuristic algorithms towards the optimum under certain conditions.
- *Machine learning* algorithms make intelligent decisions based on the data they have learned. Deep learning is a sub-field of ML that employs a layered ANN structure to learn and make smart decisions autonomously. Reinforcement learning (RL) is a field of learning in which an agent learns to make decisions through the

TABLE III
SUMMARY OF PAPERS PRESENTING HEURISTIC APPROACHES FOR VNF PLACEMENT

Ref	Objective	Environment	Algorithms	Optimization metrics: Objectives and constraints						Performance	Limitations
				Energy	Cost	Resource	Traffic	Latency	QoS		
[55]	Seek the joint optimal decision of VNF placement and CPU allocation with minimum latency and small energy consumption	SDN/NFV-based 5G network, VNF graphs	MaxZ	✓		✓	✓	✓	✓	Better than greedy and affinity based algorithms	-Two stage formulation -No network cost.
[51]	Minimize the energy consumption while satisfying latency constraints of the NS	-MEC RAN, -NS	RO, Constraint modeling	✓		✓	✓	✓		Efficient placement	-No comparison with literature, -Need to consider cost.
[62]	Reduce operational and traffic cost	NFV Architecture	SAMA	✓	✓	✓	✓			Better than existing non-coordinated approaches	-Need to consider large scale network.
[63], [76]	Find optimal VNF placement and chaining with minimum cost and latency	Distributed cloud	Eigen-decomposition heuristic		✓	✓	✓	✓		Faster than greedy	-Need to consider energy
[54]	Find the optimal placement of service function chains considering the multi-objective features of network slices.	-5G RAN network slicing, -VNF chains, -Distributed basebands	MIQCP	✓	✓	✓	✓	✓		Better than MaxSAT	-No memory size, -Not fault tolerant
[64]	Reduce the deployment cost	EPCaaS	HSD, VSD		✓	✓	✓	✓		HSD better than VSD	-No energy consumption and QoS, -Need to consider distributed DCs.
[67]	Reduce cost while meeting reliability and low latency	Cloud robotics warehousing NS	Heuristic based bin-packing approach	✓	✓	✓	✓	✓	✓	Better performance than state-of-art algorithms	
[68]	Minimize the embedding cost for joint VNF placement and traffic routing	NFV	MILP, Heuristic algorithm		✓	✓	✓	✓		Efficient for large scale network	-No energy reduction, -Need to consider latency.
[79]	Need Reduce resource consumption in a small end to-end delay	NFV, SFC	MIQCP FRAM			✓		✓	✓	Better than SRAM	-Need to consider large instances, -No energy, cost and traffic.
[81]	Address the joint VNF placement, resource allocation, and user association	MEC, SFC	MILP			✓		✓	✓	Fastest execution time for large scale network	-No comparison with other algorithms
[87]	Minimize the overall amount of computing resources	C-RAN architecture	ILP, BFIRST			✓		✓		Near to optimal solution	-Need to consider realistic case, -No energy and cost
[96]	Enhance the total throughput of slicing services	Edge cloud, Core cloud	AIA			✓	✓	✓	✓	Better than SPH in terms of throughput	-Need to consider energy consumption.
[103]	Propose a multi attribute-based QoS approach for VNF placement and service chaining	MEC	RRH, UFPH			✓	✓	✓	✓	outperforms Greedy and Random algorithms	No energy and cost.

rewards or penalties received as a result of performing one or more actions. In RL, an agent collects information about the environment, called state. Then it performs an action that moves the current setting to the next state and sends a reward to the agent. This reward reflects the measure of how the agent's action optimizes the objective function. Learning from previous experiences is a valuable capability to cope with environmental changes (e.g., change in traffic type, network configuration, etc.). Table V presents a classification of learning approaches adopted for solving complex problems in VNF placement.

TABLE IV
SUMMARY OF PAPERS PRESENTING META-HEURISTIC APPROACHES FOR VNF PLACEMENT

Ref	Objective	Environment	Algorithms	Optimization metrics: Objectives and constraint						Performance	Limitations
				Energy	Cost	Resource	Traffic	Latency	QoS		
[56]	Find the optimal VNF placement for uRLLC that reduces latency and power consumption while maximizing availability	MEC	GA	✓	✓	✓		✓		Better than exact algorithm provided by CPLEX	-No network traffic, -same number of vCPU.
[58]	Reduce the energy consumption and improve the time efficiency when placing VNFs	Large scale DC	MSGAS	✓		✓	✓	✓		Better than greedy and MSG	-Need to consider cost
[69]	Reduce the deployment cost	SDN/NFV-enabled MEC architecture	GA-based VNFPRA							Better than FF and RF algorithms	
[78]	Minimize the bandwidth dissipation and the maximum link application	NFV	Greedy NSGA-II			✓		✓		Better than NSGA-II and MOGA, Outperforms non-genetic algorithms.	-Need to consider consumed energy and cost.
[88]	Find the optimal VNF placement while considering six stringent constraints	C-RAN	Constraint programming	✓	✓	✓	✓	✓	✓	Close to expected 5G performance	-No comparison with other solutions
[74]	Maximize the served traffic demands and minimize network utilization	NFV, Edge cloud	Chain deployment algorithm			✓	✓	✓		Better than greedy and Shortest-path (SPH) heuristics	-Need to consider online problem, -No energy consumption
[97]	Enhance the mobile operator profit by considering joint decisions of VNF placement, traffic routing and resource allocation	Large scale network	MILP, MaxSR		✓	✓	✓	✓	✓	Better than Best-Fit	-Need to consider energy consumption.

IV. CNF PLACEMENT

In the next few years, network operators will tend towards cloud architectures [107] in both edge and core network [108], to increase efficiency, reliability, and scalability. Cloud-native is the process of transitioning software deployments from traditional infrastructure to software and API-enabled infrastructure to leverage automation and DevOps techniques. This transition enhances the ability to deliver services quickly and allows providers to own their customers' experience effectively. A cloud-native strategy allows providers to deploy new services rapidly with greater flexibility. Several cloud-native principles are used to deploy 5G infrastructures, including agnosticism, application resiliency, software decomposition, orchestration, and automation. Software is divided into small components using micro-services. Each component can be individually packaged using a container as a service.

However, regarding the benefits of orchestration in 5G cloud-native, finding the feasible placement of containers under CaaS architectures is still challenging. The container placement (CP) is similar to classical VM placement, where the main goal is to assign containers to suitable nodes to accomplish certain objective functions under specific resource constraints.

A. Container Placement Challenges

Containers are a sort of virtualization that works by separating system instances from user space inside a single (shared) OS kernel rather than virtualizing an entire machine as VMs do. Although this provides many benefits in communication performance between containers, it also means that all containers fight for the same resources in the system, leading to undesirable situations. A container is a unique process in the OS that does not have access to all of its resources. For example, it can only see a limited file system tree and cannot use all network interfaces; or it has limited memory allocation and disk I/O throughput. Current container service frameworks do not provide any kind of intelligent resource scheduling. Instead of taking a holistic view of all registered apps and available resources in the cloud, applications are often scheduled separately. This can lead to longer application execution times, resource wastage due to underutilized container instances, and a reduction in the number of apps that can be implemented considering the available resources, whereby the necessity of an optimal container placement approach that provides resource efficiency in a cloud environment [109]. In addition to resource utilization, the network traffic of containers should also be addressed to ensure the QoS and reduce the total energy consumption in a cloud environment [110]. As a result, virtual

TABLE V
SUMMARY OF PAPERS PRESENTING LEARNING APPROACHES FOR VNF PLACEMENT

Ref	Objective	Environment	Algorithms	Optimization metrics: Objectives and constraints						Performance	Limitations
				Energy	Cost	Resource	Traffic	Latency	QoS		
[59]	Reduce the energy consumption and performance interference	Heterogeneous	DDAP	✓	✓	✓		✓		Better than FFH and ACS	-No network traffic.
[60]	Minimize the energy consumption while achieving high availability	SFC in edge computing	PPO2, A2C	✓				✓		Better than greedy	- No network traffic, - Need to consider cost
[61]	Reduce the overall power consumption	NFV infrastructure	RL Advanced NCO	✓		✓		✓	✓	Better than FF	Need to consider cost and traffic.
[65]	Improve the quality of management and orchestration systems, Provide an optimal placement of VNFs with minimal monitoring cost.	Generic cloud system	zTORCH		✓	✓		✓		Near-optimal results compared to Instant placement (i.e., without ML)	-No energy consumption, No traffic, -Need to consider geographically distributed architecture
[66]	Reduce the total cost while ensuring QoS and low E2E latency	NFV architecture	UCB		✓	✓	✓	✓	✓	Better than greedy	-No energy consumption
[71]	Find real time VNF placement/readjustment solution that reduces SLO violation cost and latency	SFC	MAPLE	✓	✓	✓	✓	✓	✓	Outperforms k-means	
[72]	Reduce the overall cost of placing active and standby VNFs	SFC	RL		✓	✓	✓			Better than Node-Rank	-No energy consumption and traffic.
[83]	Decrease the number of network components and reduce the runtime in the Substrate Network (SN)	NFV	ILP and ML		✓	✓		✓		Better than exact algorithms	-No energy and traffic.
[90]	Place VNFs dynamically while considering SLA requirement, E2E latency and network resources	MEC, SDN	DDPG			✓		✓		Best placement compared to the *Baseline*	-Need to consider proactive state, -No energy and cost, -Unsuitable for large scale networks.
[92]	Find optimal solution of VNF-FG placement while minimizing the resource allocation and ensuring QoS requirements	NFV, Edge	DDPG-HFA, E^2D^2PG			✓			✓	E^2D^2PG outperforms FFD and DDPG-HFA	-Need to consider energy and cost.
[94]	Enable efficient SFC with optimal dynamic VNF placement		RL		✓	✓		✓	✓	Better performance than ODL scheduler	-Long time to converge in a large space
[95]	Achieve load balancing and improve long term reward and performance	NFV, SFC	EQL		✓		✓	✓	✓	Superior than ILP	-Need to consider energy consumption.
[101]	Provide a NS placement solution for non-stationary traffic conditions	Large scale infrastructure, NS	HA-DRL			✓	✓	✓		More reliable than pure DRL	

machines should be able to meet both the aggregated resource consumption and the bandwidth requirements of co-located containers. This is a challenging issue to solve due to its quadratic nature since communication between each pair of containers must be considered. Furthermore, applications should be deployed in a manner that allows them to communicate with each other with the minimum possible amount of network overhead. In this context, containers with a greater communication rate should be placed on virtual machines hosted on a single server or on servers with the shortest average

network link. Network-based placement can be efficient in terms of data center power consumption and earned revenue for cloud providers.

Poor container placement may generate a bottleneck in the cloud if VMs are substantially loaded, which impacts the response time of a particular set of tasks. For example, when certain VMs are chosen to handle container loads, some of them may already be overburdened [111]. As a result, an overhead problem arises, and the response time increases. Therefore, the VMs with greater load balancing values are more convenient for placing containers. Edge computing has the potential to expand clouds by placing virtual resources (e.g., containers) closer to data sources, allowing for faster, lower-latency applications and services. Ensuring an efficient and predictable service provisioning time presents a significant and emerging difficulty as the number of Edge-servers increases and the heterogeneity of networks linking them rises. This may result in a long provisioning time depending on the container images sizes, and the network bandwidth [112]. For instance, we frequently scale-out live video stream analytics to avoid data bursts; hence, because those application's response time are in milliseconds, waiting hundreds of seconds to provision, a new container is unacceptable [113]. Therefore, container placement models are continually being advanced for edge computing to meet the KPI performance and small latencies required by IoT services.

B. CNF Placement Proposals in Fog/Edge Computing

All previous works utilize some objectives to quantify the performance of the realized solution and evaluate the efficiency of the proposed approach to devise efficient container scheduling. Some of the well-known optimization objectives used include energy efficiency, availability, resource utilization, load balancing, scalability, cost, and makespan/latency. Other optimization objectives can be used in particular application environments or with specific data center characteristics. Readers are referred to the recent survey to examine such domain-specific optimization objectives. An optimization objective is used to measure special aspects of the solution generated by an algorithm. In some papers, the optimization objective may consist of a single objective, while others may be multi-objective, relying on the optimization required for the problem at hand. Generally, the more objectives used in the cost function for the considered optimization problem, the more complex the decision-making process becomes. Therefore, different trade-offs are typically set in place to balance the performance of the proposed algorithm and the quality of the generated solution. The following are considered the most common objectives utilized in the cost function definition of containers placement problem.

1) Resource Utilization: For resource management in the containerized cloud, new deployment models, such as fog and advanced mobile computing, have been established to make the cloud closer to the end-user [114] and services closer to the edge [115]. However, resource allocation in dynamic fog computing systems is a challenge. Authors in [116] proposed a joint optimization problem for container function placement and task provisioning. Their main objective is to optimize the number of served end-users while considering resource utilization and mobility under delay/threshold constraints. The issue is formulated as an ILP and was solved using low complex Particle-Swarm-Optimization (PSO) based meta-heuristic and Greedy heuristic algorithms. Simulation scenarios prove that the PSO-based algorithm performs near-optimal results with more sustained execution times than the Greedy Algorithm. Besides, network slicing has been introduced by 3GPP to improve the scalability of fog computing in 5G, where the E2E network in a vertical slice connected the core network to the edge devices throughout the fog nodes. Each fog node can use the harvested energy to provide pervasive computing resources anytime and anywhere. For scalable fog computing with energy harvesting, a dynamic network slicing architecture is proposed to manage the workload handled by several fog nodes located close to each other [114] and maximize the utilization of available resources.

Previous works have separately considered the placement of VMs on PMs or the placement of containers on VMs/PMs. However, this leads to over-utilized or underutilized VMs/PMs [117]. For this reason, there is growing interest in developing a container placement algorithm that considers the simultaneous use of instantiated VMs and used PMs. Cloud-native principles and technology have proven to be an effective acceleration technology in continuously building and operating the largest clouds in the world. This new technology has been selected to develop next-generation VNFs called CNFs, where network function is deployed to operate inside containers. Services are instantiated as a group of containers, which frequently leads to a high communications workload causing a degraded quality of service. Placing containers of the same service within the same server can reduce communication costs but may cause heavily imbalanced resource utilization. In this context, two phases are handled, container placement (CP) and container reassignment (CR) [118]. For the CP problem, the Worst Fit Decreasing (WFD) algorithm is proposed to provide efficient communications. For the CR problem, a reassignment algorithm named Sweap&Search is suggested to coordinate containers' distribution by migrating them among servers.

The Docker Swarm placement approach matches the containers to the available resources according to the round-robin principle without considering resource utilization of VMs or PMs. In this context, [119] proposes placing new containers on VMs while simultaneously taking into account the VM placement on PMs. The primary purpose is to reduce the number of active PMs and VMs and optimize CPU and memory usage. Therefore, the authors propose a meta-heuristic placement algorithm based on Ant Colony Optimization Best Fit (ACO-BF), which uses a fitness function that computes the percentage of wasted remaining resources in PMs and VMs. Simulation results prove the ACO-BF's performance in terms of resource usage in both PMs and VMs compared to FF and MF.

In the same way, authors in [120] have proposed a new Docker container orchestrator named Carvela. Unlike other container placement or orchestration approaches dedicated for centralized architectures [121], Carvela uses a fully

decentralized architecture and resource discovery to handle a large number of volunteer resources and avoid bottlenecks; it also employs workload placement heuristic algorithms to take the appropriate placement decision with respect to the resource (i.e., CPU, RAM) and satisfying low latency and cheap bandwidth.

Arora and Ksentini [122] propose a dynamic resource allocation and placement algorithm (DRAP) aiming to design and place a simple cloud-native network service. Their approach can help service providers to reduce their infrastructure costs. The proposed DRAP heuristic algorithm aims to reduce resource utilization while ensuring service availability. It focuses on minimizing the number of nodes needed to place CNF pods (i.e., Kubernetes pod) by adapting the vCPU allocation to each pod. The scalable vCPU allocation permits the algorithm to scale up or down the number of pods based on service availability.

Authors in [15] present a comprehensive study of container placement algorithms and scheduling models in Edge Computing. The container placement is a decision-making problem that can be formulated by graph models or multi-objective optimization models to be solved by heuristic or meta-heuristic algorithms. In [123], the container placement problem in cyber-physical systems is formulated as an ILP optimization model aiming to maximize resource utilization while ensuring high QoS. A heuristic algorithm based on Deep Learning Artificial Neural Network (ANN) is proposed to solve the ILP optimization model.

Some scheduling methods can place containers on the infrastructure with manual resource allocation that may affect the application's performance. Therefore, an automatic approach for allocating optimal CPU resources can help to improve the efficiency of containers placement. In [124], authors propose a new deep learning-based algorithm for dynamic CPU resource allocation while reducing the job completion time. This approach employs the law of diminishing marginal returns to estimate the ideal number of CPU pins for containers in order to maximize the number of concurrent jobs while maximizing performance. Experiment results, tested on a Docker-based containerized infrastructure with real workloads, prove the performance of the proposed DL algorithm in terms of decreasing the job completion time by 23% to 74% compared to static scheduling methods as First come First Serve (FCFS), Shortest Job First (SJF), Longest Job First (LJF), and Simulated Annealing (SA).

Similarly, the Docker Swarm's scheduler overlooks the resource utilization when placing containers in the cluster. In [125], authors first examine the performance interference in container placement where results show that the performance of distributed applications can be degraded when co-located containers highly consume resources. Then, they propose a new scheduler based on machine learning clustering algorithms, *K-means++* placement policy and doubling placement policy, that help to enhance performance while maintaining high resource consumption. Simulation results prove that the proposed placement strategies can improve the distributed application's performance by up to 14,5% compared to Random and Bin-packing algorithms.

2) Energy Consumption: In [126], the authors study the CP problem in terms of energy consumption; they develop a target chromosome model for optimizing energy efficiency and propose an Improved Genetic Algorithm (IGA) to find the efficient CP solution. Experiments prove that the proposed strategy is better than existing Docker Swarm strategies. Simulation results show the effectiveness and performance of IGA in terms of energy saving compared to basic GA, First-fit, and PSO algorithms.

In [127], authors propose GenPack, a new generational scheduler, for placing containers in a cloud DC to maximize energy efficiency. It learned the attributes and requirements from the system containers' runtime monitoring. Their adopted method, tested in a Docker Swarm environment, can increase energy efficiency by up to 23% compared to the built-in schedulers (i.e., Spread, binpack and random).

Container orchestration tools have emerged as an alternative to avoid the challenging problems of highly volatile workload applications and the constraints of small energy consumption and latency, though using heuristic and AI algorithms to fit the dynamic environment. In [128], a new framework called COSCO (Coupled Simulation and Container Orchestration) is developed to achieve the efficient placement of containers in fog computing environments. Besides, the authors have proposed a Gradient-based Optimization policy using back-propagation with respect to Input called GOBI to provide fast and scalable scheduling. They have also created an extended version named GOBI* to achieve QoS by providing intelligent predictions and scheduling decisions with low latency. Simulation results prove the performance of the proposed approaches (i.e., GOBI and GOBI*) in terms of reducing energy consumption, scheduling time, response time, and service level objective compared to heuristics (e.g., GA) and other learning algorithms presented in the literature.

The container placement problem can be framed into two phases (i.e., placing containers on VMs and placing VMs on PMs) while minimizing energy consumption and maximizing resource utilization. The complexity occurs when considering the heterogeneity of containers, VMs, and PMs. Instead of handling each placement separately, authors in [129] have proposed a Whale Optimization Algorithm (WOA) to solve these two placement steps as one optimization problem. The proposed algorithm efficiently minimized the overhead of creating VMs and the energy consumption compared to DGWO, TMPSO, FFD, LF, and MF.

Similarly, in [130], the initial container placement is formulated as a bi-objective optimization model aiming to minimize the power consumption while maintaining the best service performance by proposing a novel application isolation metric to quantify the overall service performance. Zhang et al. [130] propose an optimization approach called First Fit based on improving the Genetic Algorithm (FF-based-IGA) to find the efficient initial container placement solution. Simulation results prove that the proposed algorithm yields better results in terms of minimizing the energy consumption compared to conventional algorithms such as BF and FF.

In [131], a new scheduling approach based on a multi-criteria decision algorithm is proposed to place containers in

the convenient node. This approach considers three criteria: the amount of available memory, the number of containers in each node, and the number of available CPUs. The scheduling strategy aims to select the node that hosts a container by combining the Spread and the Bin Packing models to form the Technique for the Order of Prioritisation by Similarity to Ideal Solution (TOPSIS) algorithm. Simulation results prove the performance of TOPSIS in terms of reducing energy and computing time compared to Random, Spread, and Bin-packing algorithms.

Likewise, to reduce the energy consumption of service placement in cloud DC, a green container-based service aggregation is presented [132], allowing a large number of servers to be in the idle mode without impacting the quality of experience. The problem has been formulated as an optimization approach to minimize the total energy consumption with respect to service execution time. In this way, the authors propose an online learning-based technique based on Bayesian Optimization (BO) that can handle measurement noises encountered during workload characterization for containerized services. This algorithm is named Energy-Aware Service consolidation using Bayesian optimization (EASY). The experiments executed in the docker swarm environment prove the effectiveness of EASY in reducing the total energy consumption and bandwidth overhead compared to FFD and BF. However, as the reduction of active nodes makes them heavy loaded, the average response time is greater than baseline methods.

The goal of resource allocation in container-based clouds is to reduce total energy consumption by properly assigning resources (such as CPU and memory) to applications without overloading the PMs. Hence, regarding the elastic nature of containers, a cloud provider must distribute appropriate resources as soon as a new request arises, and this is named online Resource Allocation in Container-based clouds (RAC) problem. It is challenging because of its two-phase strategy (i.e., placing containers in VMs and placing VMs in PMs). Previous studies learn a one-stage allocation policy for allocating containers to VMs. In contrast, the assignment of VMs to PMs is manually performed. In [133], authors present a novel Cooperative Coevolutionary Genetic Programming (CCGP) hyper-heuristic algorithm to address the RAC problem by learning the workload pattern and generating allocation rules for the two levels. Simulation results prove the performance of CCGP rules in terms of improving energy efficiency compared to the sub&Just-Fit/FF rule.

3) Network Traffic: The containers implemented in an application are located in several PMs to ensure high parallel performance. The CNF placement has a significant impact on the network traffic and the containerized data center performance. Unlike the existing CNF placement solutions that do not consider the traffic pattern of containers, authors in [112] propose a new placement approach based on network traffic correlation named "Blender" considering the traffic between containers as a Zipf distribution. The blender approach offers two valuable benefits: (i) it reduces inter-block traffic by placing containers that often communicate in the same block. (ii) it performs efficient load balancing by grouping blocks depending on the types of required resources and dispatching them over several PMs. Simulation results prove the high performance of the Blender solution compared to SBP and CA-WFD in terms of reducing communication traffic.

The critical challenge in container cluster (CC) provisioning is the efficient placement of containers while considering inter-container traffic. This challenge is further complicated when the clusters of containers are provisioned online. Hence, authors in [134] propose an online placement algorithm to dynamically assign the container to a zone with free capacity while considering the inter-container traffic. This online placement design involves a one-shot algorithm that identifies the optimal placement for the current CC and an online algorithm framework that makes on-spot decisions upon the arrival of CC requests and based on resource prices. An exhaustive sampling and ST rounding techniques were applied to reduce the complexity degree of the one-shot CC placement problem and find efficient solutions. In addition, compact-exponential and primal-dual online methods are exploited to ensure a good competitive ratio.

Monitoring inter-application traffic properly without instrumenting the application, required to dynamically determine the appropriate container placement, is difficult to achieve. In [135], authors propose an effective black-box monitoring strategy for identifying and constructing a weighted communication graph of cooperating processes in a distributed system that can be accessed for a variety of reasons, including adaptive placement.

In [136], authors propose an Availability-assured Buffered-layer Prioritized scheduler (ABP) to minimize network traffic and reduce the latency of scaling services in Docker Swarm. They adopt a heuristic algorithm named Dominant Resource Fairness to run this scheduler. Experiments show that the ABP scheduler significantly improves service creation and deployment in the Docker swarm environment.

Distributed cloud is a vital technology for 5G networks and is emerging as an alternative for managing latency-sensitive and traffic-intensive applications. Placing containers on the edge cloud enables applications to be located closer to end-users and traffic sources, which will result in reducing latency and network traffic. In this context, authors in [137] propose a two-level approach to tackle the traffic and latency-aware container placement optimization problem in a distributed cloud. They use an ILP model to solve the first step of placing containers in DCs, considering the average cost, resource utilization, and acceptance ratio as key performance metrics. For the second step of placing containers in servers, a traffic-aware heuristic algorithm is proposed. Results prove the performance of the proposed heuristic approach in terms of reducing all traffic metrics compared to conventional bin-packing heuristics (i.e., FFD, BFD).

4) Response Time, Execution Time, Communication Cost: Nowadays, telecom operators tend to deploy their 5G services in the form of containers in their large-scale data centers. Each service includes multiple modules that are instantiated as a group of containers, where containers owned by the same service commonly need to communicate with each other to provide the required service [138] leading to cumbersome inter-server communication and service performance

degradation. The communication cost can be significantly decreased if these containers are placed on the same server. Nevertheless, containers belonging to the same service are typically exhaustive on the same resource. For example, containers of data transfer applications are network I/O intensive [139], thus resulting in unbalanced resource usage, but this can have a positive impact on availability, response time, and system throughput. However, reducing the communication cost while maintaining a balanced use of resources is challenging. In this context, for this conflicted goal, authors in [118] handle the problem in two phases: container placement and container reassignment. The first one aims to place a set of containers on DCs to minimize resource utilization while reducing the communication cost by using a Worst Fit Decreasing (WFD) algorithm. The second phase of container reassignment attempts to optimize a given container placement by migrating containers between servers. The authors have proposed a two-stage approach, Sweep & Search, that first tackles overloaded servers and then optimizes the targets using local search techniques. Simulation results prove the performance of the proposed algorithms in terms of low cost, high throughput, and balanced resource utilization compared to the state-of-art algorithms.

The optimal container placement in volatile fog nodes (i.e., CPU, communication fabric, memory, storage) can be ensured by reducing costs and guaranteeing the customer's QoS (i.e., response time and isolation). In [140], a two-phase partition-based optimization approach is proposed to improve the service availability and QoS satisfaction through initially matching applications to fog device communities and then placing application services transitively on fog devices. However, in this case, the inter-container communication was neglected, and few are the papers [118], [141] that addressed the impact of network communication on isolation and QoS in fog computing. In [141], authors have proposed an optimal genetic algorithm for container placement aiming to reduce the response time while considering the heterogeneity of fog nodes and inter-container network communication as well as the isolation requirements for applications deployed on fog computing networks. For inter-container communication, three modes have been deployed (i.e., host mode, overlay mode, and RDMA). Results prove the performance of the proposed GA algorithm compared to greedy and ILP in terms of isolation and significant response time reduction when using a greater number of RDMA-enabled fog nodes.

The convenient placement of containers on VMs can help to optimize resource utilization in cloud environments. However, the bad placement may result in a bottleneck in the cloud if VMs are highly congested, which can adversely impact the response time of a given set of tasks. In [142], authors propose an Ant Colony Optimization (ACO) algorithm to reduce the overall makespan of tasks, thus leading to reduce the response time of applications. The typical ACO tends to schedule tasks to the most used node, which can cause an overload issue if the node is carrying a large load. The drawback of ACO is its disregard of resource utilization and energy efficiency. Considering these challenges, authors in [143] have proposed a Modified ACO (MACO) for container placement to optimize the response time and improve the scheduling decision while taking into account resource utilization, throughput, and energy consumption. Simulation results show that MACO outperforms the First Come First Serve algorithm (FCFS) in response time and throughput.

Similarly, authors in [144] have proposed a container-based task scheduling using a hybrid bacteria foraging optimization (HBFA) algorithm to minimize the execution time and increase the resource utilization in an edge computing environment. The proposed HBFA yields better scheduling results than BF and GA.

Current serverless platforms have multiple constraints in terms of supporting data-centric distributed computing, which are compounded by the operational features of underlying edge systems, particularly when it comes to function placement decisions. Among constraints, the high latencies incurred by the distance between nodes in edge computing infrastructures. Therefore, the inter-node proximity and bandwidth must be taken into account. In [145], a new container scheduling system called Skippy is proposed to provide an efficient placement of edge functions by considering the scheduling limitations of the application's data flow, network topology, proximity, and available compute capabilities. Experiments prove the performance of Skippy in terms of reducing execution time, cloud execution cost, and uplink usage.

5G networks, driven by NFV, promise to enable a wide range of services from various market segments (e.g., Smart Cities, smart homes, Automotive, etc.). Services must be connected in a precise order to properly benefit from NFV, forming a Service Function Chain (SFC). The majority of existing works handle the placement of VNF-based service chains, with the target to find the optimal placement while minimizing the end-to-end latency and maximizing the resource utilization [79]. Many optimization models based on ILP [146], [147] have been proposed to facilitate the SFC orchestration through deciding whether to migrate or replace VNFs while reducing the SFC latencies. Nonetheless, few papers considered the latency-aware container-based SFC chain in fog computing. In [148], authors propose an SFC controller, as an extension of Kubernetes scheduling features, to optimize the placement of container-based SFC while optimizing resource provisioning and minimizing the E2E latency.

Similarly, assuming that the RAN and core functions are deployed as CNFs in the data centers. Users of Service Function Request (SFR) connect to remote radio heads (RRHs) in order to receive service. In [149], the authors propose a mathematical model for CNF placement and resource allocation of an ORAN enabled 5G network with the objective to minimize the End-to-End delay of the data plane. They consider two cases, the first one where the traffic of an SFR crosses a single path through CNFs of its chain. They formulate this scenario as a non-linear mixed inter programming approach and then its been converted to a liner problem after some reformulations, but its non-trivial. The second one where SFR traffic can traverse multiples paths of CNFs. In this scenario, the authors propose a gradient based minimum delay algorithm (GBMD). Simulation results prove that GMDB help to reduce End-to-End delay by 90% compared to single path.

Several placement techniques based on deep reinforcement learning (DRL) have been proposed in cloud or edge computing environments, but they are not suitable for distributed architectures. The task of forming efficient DRL agents involves a lot of training data, and their procurement is expensive. The centralized DRL-based strategies suffer from poor scalability and are therefore unable to solve placement issues. Many IoT applications are created using Directed Acyclic Graphs (DAGs) with different topologies. Meeting the requirements of DAG-based IoT applications leads to further constraints and makes the placement problem more complex. To address these issues, authors in [150] propose a distributed DRL approach named X-DDRL that aims to solve the placement challenge of DAG-based IoT applications while minimizing the energy consumption and the execution time. For training the distributed brokers, an actor-critic-based distributed application placement technique named IMPALA (IMPortance weighted Actor-Learner Architectures) is proposed to achieve timely and efficient application placement decisions. IMPALA framework can minimize the agent's exploration costs and provides rapid convergence to optimal solutions.

In [151], to manage CNFs efficiently, authors propose a deep Q-network-based CNF placement algorithm (DQN-CNFPA), which jointly reduces the cost of launching and operating CNFs on edge clouds and the back-haul control traffic overhead, and maximize the number of served requests at each time. Simulation results show that DQN-CNFPA can distribute CNFs in a manner that takes into account fluctuations in service demand. The proposed algorithm can minimize the cost per hour by up to 11.2% compared to a scheme that does not take into account fluctuations in service demand.

5) QoS: In smart applications (e.g., smart cities and smart homes), the big data workflow is based on various sensors and video streams where AI and feature extraction techniques are performed. The captured information is stored in DB containers. These containers need to be placed on Edge, Fog, or Cloud infrastructures while addressing the QoS requirements. Open source solutions such as Docker or Kubernetes can orchestrate containers in edge and fog computing, but the decision on where to place a software instance considering major QoS metrics (i.e., throughput, latency, power consumption, CPU utilization, cost). In [152], a stochastic approach for DB container placement based on Markov Decision Process (MDP) is proposed to (i) dynamically enhance the automation based on new QoS attributes, (ii) build utility functions that provide the reward values and help to find the optimal solution of decision making, (iii) ranking deployment infrastructures based on rewards to get the QoS success score. The authors also propose a new architecture that automates the whole process. The author's experiments were based on 25 infrastructures and 8 QoS attributes. Simulations prove that MDB is better than Analytic Hierarchy Process (AHP) method in terms of QoS violations, where no violation is faced for MDB, in contrast to AHF, where QoS violation is omnipresent in all workload scenarios.

In a multi-level environment of cloud/fog/edge, the network has a crucial role as it serves as a communication link between all of the system's participants; as a result, its performance has an impact on the whole system. Therefore, the network should be considered while making all placement decisions in order to meet the QoS performance. In [153], the QoS assurance techniques in fog computing are categorized into service/resource management, communication management, and application management. Accordingly, the container QoS can be satisfied by managing network and storage workloads. In [154], authors propose a container management strategy named CONtrol to balance the bandwidth between storage traffic and application traffic over a hyper-converged architecture. The primary objective of CONtrol is to manage the storage traffic when scheduling containers while sustaining the QoS of the container network. CONtrol aims to make dynamic placement decisions over bandwidth redistribution across diverse workloads using a proportional-integral-derivative controller. In the same way, a new scheduler module (i.e., an extension of Kubernetes scheduler) is proposed to make placement decisions based on network status [155]. Here, the authors develop a network-aware scheduling algorithm named IPerf aiming to compute the estimation time for job completion where the system rejects the applications that do not meet the deadline.

6) Security: The majority of container placement research focuses on dealing with resource utilization, energy consumption, cost, response time, etc. However, few works consider the security problems. In this subsection, we cite some security strategies treated in previous works, such as avoiding co-location attacks, improving user isolation, and identifying vulnerabilities. The various co-resident attacks present a significant challenge to cloud providers and tenants. Each type of co-resident attack needs significant changes in hardware, host systems, container engines, and system configurations. On the other hand, as the overall co-residence is unknown (and increasing), it is difficult to fix the software and hardware to address future attacks. The container deployment approach offers a straightforward and effective way to influence the likelihood of co-residency.

In [156], authors confirm that containers placed in VMs are susceptible to co-residency attacks. The co-residency detection can tolerate background noise with a 70% success rate, as long as it does not surpass the hardware capacity. Their analyses show that any change in architecture and orchestrator can reduce detection fidelity by up to 10%. Therefore, cloud customers should not rely on orchestration platforms to satisfy sufficient protection against co-residency attacks.

In [157], authors propose a Secure Container Deployment Strategy named SecCDS based on Genetic Algorithm (GA) to cope with co-resident attacks in container clouds. They carefully orchestrate the placement and migration of containers to dissociate attackers and victims on various PMs. The GA-based strategy aims to overcome the problem of selecting the target to which the containers migrate. Meanwhile, to increase the convergence time of GA, a Simulated Annealing (SA) algorithm is proposed by performing a strong neighborhood search for each unit in GA. Simulations prove that the proposed approach yields better results in terms of minimizing the co-residency attacks with negligible effect on system performance and workload compared to classic strategies such

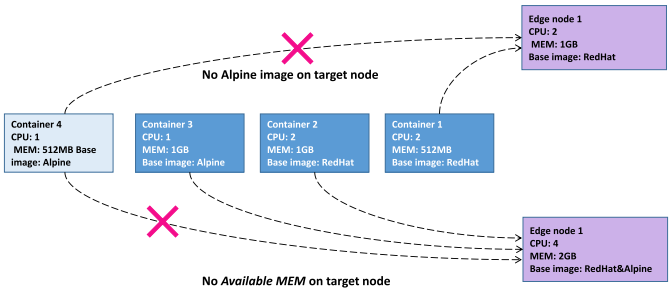

Fig. 15. Queuing approach.

as Previous Selected Server First (PSSF), Random, Most and Least strategies.

C. Classification of Container Placement Approaches

The container placement approaches can be categorized into two types: queuing and concurrent placement. The queuing approach can be defined as a FIFO or priority-based approach where the CP decision is performed on a container by container [158], [159], [160], [161]. The container-by-container placement approach has the benefit of making parallel decisions on a distributed architecture; however, it also has critical restrictions for ensuring efficient placements of all concurrent tasks as it does not have a holistic view of pending containers. In general, an optimal decision is difficult to achieve since, in queuing models, the initial placement decision is taken by the first container in the queue regardless of the other remaining containers in the queue. However, if the specifications of all requests are known beforehand, some scheduling rules for smart container placement can be developed based on machine learning algorithms to predict all incoming requests [162], [163].

The concurrent approach is defined as a batch processing concept where computing requests are first gathered, and then a placement decision is made [164], [165]. Here, the scheduler has a complete view of all workloads, where all containers can be placed in convenient VMs. However, the concurrent scheduling strategy may be highly complex, as the problem is often formulated as integer programming or mixed-integer programming, which may impact the QoS. Also, the batching time needed for intermittent requests can delay the placement as the allocation task waits for a certain time to serve multiple requests.

Figures 15 and 16 depict an example of the two approaches (i.e., queuing and concurrent) for four concurrent requests. As depicted in Figure 15, for queuing model, container four can not be placed on any node due to the lack of resources after placing the three other containers. On the other hand, as seen in Figure 16, the container scheduling is optimized when using the concurrent approach.

The strategies used for container placement are: Spread (tries to place the containers evenly on available nodes), Binpack (place containers on the most-loaded host that still has enough resources to run the given containers), and custom. Similar to VNF placement, the surveyed scheduling algorithms adopted for container placement are classified into three categories: Heuristics, meta-heuristics, and machine learning.

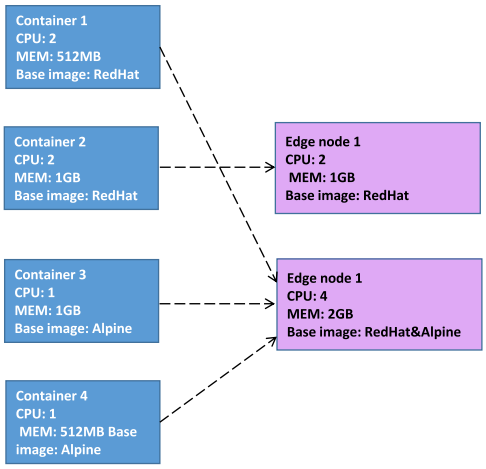

Fig. 16. Concurrent approach.

The majority of the reviewed techniques use some heuristics to get approximate solutions to the problem, as shown in Table VI. Heuristic algorithms are typically of low complexity and generate a suitable program in a reasonable time.

Meta-heuristics are a flexible and popular class of population-based optimization algorithms inspired by the intelligent processes and behaviors of nature. Meta-heuristics are widely used to solve optimization problems in several disciplines. Two important features of these algorithms are a selection of the fittest and adaptability to the environment. Table VII provides a classification of meta-heuristic algorithms used for container placement.

Machine learning is an active field of research with a lot of success in various applications, and it is very promising for container placement. ML algorithms are successful because of the availability of big data to train the model. Compared with other heuristics, one can benefit from machine learning techniques to improve solution accuracy and effectiveness by making intelligent scheduling decisions. We present in Table VIII a classification based on performance metrics and machine learning algorithms.

V. Insights & Future Directions

A. Synthesis

For a telecom operator, providers are converging towards containerized architectures. Therefore, instead of focusing on the placement of VNFs, the placement and orchestration of containers in edge and fog computing must be taken into account to provide services with good performance and high QoS. For VNF/CNF placement, the optimization approaches were classified into six objective functions, where the major objective for telecom operators is reducing the latency while considering energy consumption, cost, resource utilization, security and QoS.

However, it is a bit difficult to optimize these several conflicting objectives simultaneously. The VNF/container placement problem is defined under various parameters regarding the type of objective functions, the constraints, and the environment. These parameters differ from one scheme to another, and choosing the problem setting depends heavily on the

TABLE VI
SUMMARY OF PAPERS PRESENTING HEURISTIC APPROACHES FOR CONTAINER PLACEMENT

Ref	Objective	Environment	Algorithms	Optimization metrics: Objectives and constraints						Performance	Limitations
				Energy	Cost	Resource	Traffic	Latency	QoS		
[118]	Balance resource utilization and minimize communication cost	Large scale DCs	WFD, Sweep & Search		✓	✓		✓		-WFD outperforms Bin-packing and Random, Sweep&Search better than Greedy	-Need to include traffic and energy consumption.
[120]	Handle the large number of volunteer resources and avoid bottlenecks with low latency and bandwidth	Decentralized architecture	Caravela with heuristic algo-rithms	✓		✓		✓		Better than random ap-proach	No traffic.
[122]	Reduce the resource utilization while ensuring the service availability	NFV	DRAP		✓	✓		✓	✓	Better than Gurobi algorithm	-Simple CNFs, -No energy and traffic
[130]	Reduce the power consumption while ensuring best service performance	NFV	FF-based-IGA	✓	✓	✓		✓	✓	Better than FF and BF	-Homogeneous VMs and PMs, -Static placement, -Need to consider network traffic interference.
[133]	Reduce the total energy consumption by properly assigning resources to applications without overloading the PMs	Container-based clouds	CCGP	✓		✓		✓	✓	Better than sub&Just-Fit/FF	-Need to consider the network traffic.
[112]	Reduce inter-block traffic and provide efficient load balancing between resources	containerized DCs	Blender			✓	✓	✓	✓	Better than SBP and CA-WFD	-Need to consider migration.
[136]	minimize network traffic and reduce the latency of scaling services	Cloud	Dominant Resource Fairness, Abp scheduler			✓	✓	✓	✓	Better than default scheduler	-Need to consider dynamic placement.
[137]	Address the traffic and latency-aware container placement optimization problem	Distributed cloud	Traffic-aware heuristic		✓	✓	✓	✓	✓	Better than FFD and BFD	-Need to consider energy consumption.
[140]	Improve the service availability and QoS satisfaction	Fog comput-ing	Two-phase heuristic			✓		✓	✓	Outperforms ILP in terms of response time	-Need to consider total service cost and network usage.
[144]	Reduce the execution time and increase the resource utilization	Edge cloud	HBFA		✓	✓		✓		Better than BF and GA	-Need to consider energy consumption and resource estimation.
[145]	Provide an optimal placement of edge functions, by considering the scheduling limitations of application's data flow, network topology, proximity and available compute capabilities	Edge cloud	Skippy		✓	✓	✓	✓			-Need to consider auto-scaling and workload migration.
[152]	Provide automatic container placement considering QoS metrics	Edge and Fog	MDP		✓	✓	✓	✓	✓	Better than AHP	-Need to consider energy consumption and migration.
[155]	Enhance the QoS and reduce the time of job completion	Fog comput-ing	IPerf		✓	✓	✓	✓	✓	Better than AHP	-Need to consider energy consumption and migration.

context and the scope of dealing with virtual resource placement. For example, if high energy consumption is the most critical issue in a DC, it would be a primary objective in the problem formulation, and the other less critical issues can be ignored or considered as constraints. Therefore, based on the literature review, this mono-objective problem can be

TABLE VII
SUMMARY OF PAPERS PRESENTING META-HEURISTIC APPROACHES FOR CONTAINER PLACEMENT

Ref	Objective	Environment	Algorithms	Optimization metrics: Objectives and constraints						Performance	Limitations
				Energy	Cost	Resource	Traffic	Latency	QoS		
[116]	Optimize the number of served users while considering the resources utilization and mobility under delay/threshold constraints	Fog computing	PSO			✓		✓	✓	Better than greedy and near to optimal	-Need to consider traffic congestion.
[119]	Minimize the number of active PMs and VMs and optimize resource utilization in terms of CPU and memory usage	CaaS cloud	ACO-BF			✓		✓		Better than BF and MF	-Need to consider traffic.
[126]	Optimize energy efficiency	CaaS	IGA	✓		✓		✓		Better than basic GA and PSO	-Static container placement, -Need to consider network traffic and cost.
[129]	Reduce the energy consumption and the overhead of creating VMs	CaaS cloud	WOA	✓		✓		✓		Better than DGWO, TMPSO, FFD, LF and MF	-Static container placement, -Need to consider network traffic, migration time and SLA.
[134]	Place dynamically container to a zone with free capacity while considering the inter-container traffic	Cloud container cluster	Online algorithm		✓	✓	✓	✓	✓	Better than greedy and rounding algorithms	-Need to consider energy consumption.
[141]	Reduce the response time while considering the heterogeneity of fog nodes, inter-container network and the isolation	Fog computing	GA		✓	✓	✓	✓	✓	Better than greedy and ILP	
[142]	reduce the overall makespan of tasks, thus leading to minimize the response time of applications	Fog computing	ACO					✓	✓	Better than random and FF	-Need to consider resource utilization and energy efficiency.
[143]	Optimize the response time and improve the scheduling decision while taking into account resource utilization, throughput and energy consumption	CaaS	MACO	✓	✓	✓		✓	✓	Outperforms FCFS	-Need to consider SLA.

solved using heuristic or deterministic algorithms. However, if there are more than two or three conflicting goals, the problem is considered a multi-objective optimization approach. Several approaches have been proposed to solve this type of problem, based on heuristic and meta-heuristic algorithms. Besides, machine learning algorithms are well recommended for large data centers and distributed architectures to find optimal solutions for complex problems.

This paper classifies placement techniques into three categories based on the adopted optimization algorithms. We explore the optimization objectives to evaluate the performance of the generated scheduling. We have described and evaluated existing strategies for each type based on their key performance indicators to identify their advantages and limitations. The future expansion of container technology will create major changing standards, requiring the development of new orchestration solutions, placement, scheduling, and resource management. Emerging technologies like Edge/Fog computing and micro-services provide new ways to provide real-time schedulers that are sensitive to energy, communication, inter-container traffic, and security variations for these environments.

From the studied papers and industry perspectives, we answer the question of why using CNFs instead of VNFs. CNFs are lighter and more flexible than VNFs. CNFs can operate in a microservices architecture that provides a dynamic, flexible, and scalable architecture for 5G. VNFs in the telecom clouds and contemporary CNFs need to communicate

TABLE VIII
SUMMARY OF PAPERS PRESENTING LEARNING APPROACHES FOR CONTAINER PLACEMENT

Ref	Objective	Environment	Algorithms	Optimization metrics: Objectives and constraints						Performance	Limitations
				Energy	Cost	Resource	Traffic	Latency	QoS		
[123]	Maximize the resource utilization while ensuring high QoS	Cyber physical systems	ANN			✓		✓	✓	Very efficient in saving resources	-No comparison with other algorithms, -Need to consider traffic management.
[125]	Enhance performance with high resource utilization	Docker swarm and Amazon ECS	K-means+		✓	✓		✓	✓	Better than random and bin packing	-No network traffic.
[124]	Provide dynamic CPU resource allocation while minimizing the job completion time and increasing the applications performance	Docker-based containerized infrastructure	Deep learning			✓		✓	✓	Better than static scheduling (i.e., FCFS, SJF. LJF, SA)	-Need to consider disk IO and memory, -No traffic.
[126]	Maximize energy efficiency	Docker swarm environment	GenPack	✓	✓	✓		✓	✓	Better than Random, Spread and Binpack scheduling	-Need to consider network and disk intensive.
[149]	Reduce the End-to-End delay for data plane	ORAN 5G Network	GBMD			✓		✓		Better than single path	
[132]	Reduce the total energy consumption without impacting the QoE	Cloud	EASY	✓	✓	✓		✓	✓	Better than FFD and BF	-The average response time is more than baseline -Need to consider the overload.
[150]	Reduce the energy consumption and the execution time	Heterogeneous fog computing	X-DDRL	✓		✓		✓		Better than other DRL-based techniques	-Need to consider the dynamic change of transmission power.
[151]	Reduce the cost of launching and operating CNFs on edge clouds and the back-haul control traffic overhead	Private 5G network	DQN-CNFPA	✓	✓	✓		✓		Better than other DRL-based techniques	Need to consider RAM energy

with very high data rates and low latencies. Virtualization of network functions (VNFs) running on software-defined networks has increased deployment agility while enabling low-cost, standardized servers. However, the weight of VMs limits the effectiveness of VNFs for large-scale 5G, making it difficult for scalability. CNFs have further enhanced the density of functions per host, leading to more stringent performance requirements. Serverless CNFs are now driving the deployment of fault-tolerant Cloud-Native 5G. CNF delivers higher resource efficiency by implementing more services on the same server using the native microservices structure and the containerization concept. It offers better resiliency and availability as microservices are distributed across multiple servers and machines with a shared processing load. CNF helps to minimize network downtime by using continuous upgrades of microservices. Cloud-native provides greater development speed for scaling the network using the Kubernetes orchestrator.

By analyzing Tables III, IV, V, VI, VII, and VIII, some interesting trends appear. For example, regarding the optimization metric for VNF placement, fewer papers use Energy as a metric. The same holds when we analyze the optimization metric for container placement. In this case, the Traffic seems to be under-investigated. However, the Energy and Traffic metrics are less treated in CNF placement compared to VNF placement. The Resource utilization and Latency are the most treated metrics, which makes sense as improving resource utilization help to reduce latency. The QoS feature is more handled in CNF placement regarding the requirements of high quality of service and high performance. In terms of approaches, most papers use various algorithms to solve complex problems, using heuristic/metaheuristic algorithms to solve the multi-objective system. For better results, Deep learning algorithms are employed as they are faster and more efficient. It seems difficult to identify the best solution for VNF or CNF placement as the initial conditions are different, but we can confirm that DL algorithms provide best results than heuristic, meta-heuristic and greedy algorithms. We present in Figure 17 a taxonomy of VNF/CNF placement solutions.

B. Future Challenges

5G mobile broadband network operators must meet large-scale, complex, dynamic, and highly distributed infrastructure requirements. They must roll out and operate thousands of

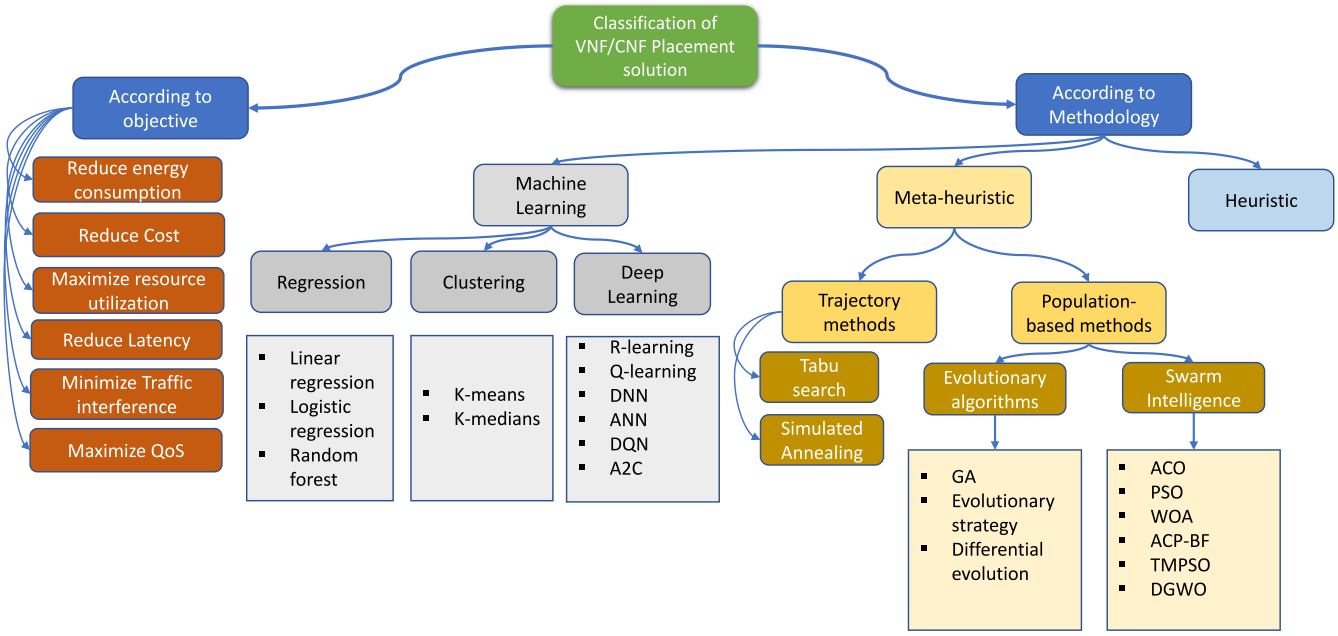

Fig. 17. Taxonomy of virtual resource placement techniques.

radio antennas and networks while simultaneously handling software applications on the access layer, aggregation layer, and central data centers. In addition, they have to satisfy stringent latency and network performance specifications for their applications and infrastructure. Furthermore, operators require the flexibility to dynamically relocate services to enhance network performance, reduce latency and minimize operating costs.

As a result, 5G architectures must be service-based on hundreds and thousands of network services in the form of VNFs or CNFs that are spread across geographically distributed remote environments. Kubernetes overcomes a part of this challenge through its ability to orchestrate and manage CNFs. However, it suffers from several limitations in handling 5G services in distributed locations with stringent latency and performance requirements.

By 2024, 5G is estimated to handle 25% of all mobile traffic, leading to faster adoption and deployment of CNFs. However, the overwhelming majority of current networks will continue to be VNFs-based. The fact is that VNFs and CNFs will have to coexist. Therefore, telecom operators must maintain two separate management stacks to run 5G and the legacy networks, which generates additional operational burden and cost. Considering the number of sites to be managed, they are faced with a proliferation of the control plane and inefficiency of siloed management, hence the need to address the coexistence of VNF and CNF in the same architecture.

Some research efforts have been dedicated to develop scheduling models for container placement and migration in fog-edge computing. Container migration is a specific type of container placement, that is rarely covered by advanced scheduling models. Furthermore, designing a security-conscious scheduler to prevent security threats connected with containers when deployed across cloud infrastructures might be an interesting subject of future research. The sustained success and attraction of deep learning algorithms can help to build smart scheduling policies by predicting future workloads and capacities for container placement and consolidation, load balancing, and resource provisioning. More fine-grained system counters are expected to control and gather metadata for prediction and decision-making in order to make the most significant use of deep learning algorithms. Furthermore, to handle energy usage, SLAs, and QoS, multi-objective holistic management container scheduling strategies must be explored.

Currently, most of the scheduling models rely on a centralized decision-making system. These centralized servers will become overloaded as edge computing continues in its growth trajectory. This will necessitate more clusters, further increasing the workload for the orchestrator in managing many master nodes. This can lead to increased latency because in a centralized scheduling system, additional time is taken to upload system states and wait for dispatch decisions. Decentralizing the scheduling decisions to multiple edge servers is inevitable. There are few efforts in this area. Therefore, in our future work, we will compare a centralized scheduling system to a decentralized one for service placement policies. We will also propose a multi-objective decision making approach for virtual resource placement in a 5G architecture based on VNFs and CNFs that can be solved based on a reinforcement optimization algorithm.

VI. Conclusion

MEC and NFV have emerged as promising technologies to deliver low latency slicing services in the 5G communication network. However, because of the large number of nodes and links in today's Data Centers, NFV raises a number of challenges where the most significant one is the difficulty of placing VNFs in physical networks and the inter-dependency

among VNFs composing a given network service. Several contributions have been made to address these challenges in a static or dynamic manner. This paper provides a classification of the existing virtual resource (VNF or Container) placement methods and algorithms. The optimization problem may consist of a single-objective or a multi-objective depending on the performance metrics. We categorize the scheduling techniques into three folds: heuristics, meta-heuristics, and machine learning algorithms. Besides, we identify the performance metrics, advantages, and limitations for each category. We also highlight the convergence towards cloud-native infrastructures.

REFERENCES

[1] E. Ahvar, S. Ahvar, S. M. Raza, J. M. S. Vilchez, and G. M. Lee, "Next generation of SDN in cloud fog for 5G and beyond-enabled applications: Opportunities and challenges," *Network*, vol. 1, no. 1, pp. 28–49, 2021. [Online]. Available: https://www.mdpi.com/2673-8732/1/1/4

[2] F. Rinaldi, A. Raschellà, and S. Pizzi, "5G NR system design: A concise survey of key features and capabilities," *Wireless Netw.*, vol. 27, no. 8, pp. 5173–5188, Nov. 2021. [Online]. Available: https://doi.org/10.1007/s11276-021-02811-y

[3] D. Wypiór, M. Klinkowski, and I. Michalski, "Open RAN & mdash radio access network evolution, benefits and market trends," *Appl. Sci.*, vol. 12, no. 1, p. 408, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/1/408

[4] F. Auer, V. Lenarduzzi, M. Felderer, and D. Taibi, "From monolithic systems to microservices: An assessment framework," *Inf. Softw. Technol.*, vol. 137, Sep. 2021, Art. no. 106600. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950584921000793

[5] H. Yang, T. So, and Y. Xu, "Chapter 12-5G network slicing," in *5G NR and Enhancements*, J. Shen, Z. Du, Z. Zhang, N. Yang, and H. Tang, Eds. Amsterdam, The Netherlands: Elsevier, 2022, pp. 621–639. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B978032391060600012X

[6] L. Guevara and F. A. Cheein, "The role of 5G technologies: Challenges in smart cities and intelligent transportation systems," *Sustainability*, vol. 12, no. 16, p. 6469, 2020. [Online]. Available: https://www.mdpi.com/2071-1050/12/16/6469

[7] L. Heng, G. Yin, and X. Zhao, "Energy aware cloud-edge service placement approaches in the Internet of Things communications," *Int. J. Commun. Syst.*, vol. 35, no. 1, 2022, Art. no. e4899.

[8] B. Costa, J. Bachiega, L. R. de Carvalho, and A. P. Araujo, "Orchestration in fog computing: A comprehensive survey," *ACM Comput. Surveys*, vol. 55, no. 2, pp. 1–34, 2022.

[9] W. Attaoui and E. Sabir, "Multi-criteria virtual machine placement in cloud computing environments: A literature review," 2018, *arXiv:1802.05113*.

[10] W. Attaoui, E. Sabir, H. Elbiaze, and M. Sadik, "Multi objective decision making for virtual machine placement in cloud computing," in *Proc. Int. Conf. Netw. Games Control Optim.*, 2021, pp. 154–166.

[11] F. Schardong, I. Nunes, and A. Schaeffer-Filho, "NFV resource allocation: A systematic review and taxonomy of vnf forwarding graph embedding," *Comput. Netw.*, vol. 185, Feb. 2021, Art. no. 107726. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128620313189

[12] A. Alashaikh, E. Alanazi, and A. Al-Fuqaha, "A survey on the use of preferences for virtual machine placement in cloud data centers," *ACM Comput. Surveys*, vol. 54, no. 5, pp. 1–39, May 2021. [Online]. Available: https://doi.org/10.1145/3450517

[13] X. Li and C. Qian, "A survey of network function placement," in *Proc. 13th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, 2016, pp. 948–953.

[14] S. Demirci and S. Sagiroglu, "Optimal placement of virtual network functions in software defined networks: A survey," *J. Netw. Comput. Appl.*, vol. 147, Dec. 2019, Art. no. 102424. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804519302760

[15] O. Oleghe, "Container placement and migration in edge computing: Concept and scheduling models," *IEEE Access*, vol. 9, pp. 68028–68043, 2021.

[16] R. Buyya and S. N. Srirama, "Management and orchestration of network slices in G, fog, edge, and clouds," in *Fog and Edge Computing: Principles and Paradigms*. Hoboken, NJ, USA: Wiley, 2019, pp. 79–101.

[17] B. Sonkoly, J. Czentye, M. Szalay, B. Németh, and L. Toka, "Survey on placement methods in the edge and beyond," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 4, pp. 2590–2629, 4th Quart., 2021.

[18] Linux Foundation. "Open Network Automation Platform." Accessed: 2022. [Online]. Available: https://www.onap.org/

[19] G. Lavado. "5G Projects Building Strong Use Cases for Open Source MANO NFV." Accessed: 2022. [Online]. Available: https://osm.etsi.org/

[20] N. Nikaein, C.-Y. Chang, and K. Alexandris, "Mosaic5G: Agile and flexible service platforms for 5G research," *SIGCOMM Comput. Commun. Rev.*, vol. 48, no. 3, pp. 29–34, Sep. 2018. [Online]. Available: https://doi.org/10.1145/3276799.3276803

[21] "Open5GCore." Accessed: 2021. [Online]. Available: https://www.open5gcore.org/

[22] P. M. Mell, *The NIST Definition of Cloud Computing*, document SP-800, Nat. Inst. Sci. Technol., Gaithersburg, MD, USA, 2011.

[23] "What is Cloud Computing." Accessed: 2022. [Online]. Available: https://www.ibm.com/cloud/learn/cloud-computing

[24] M. Saraswat and R. Tripathi, "Cloud computing: Analysis of top 5 CSPs in SaaS, PaaS and IaaS platforms," in *Proc. 9th Int. Conf. Syst. Model. Adv. Res. Trends (SMART)*, 2020, pp. 300–305.

[25] M. Stieninger and D. Nedbal, "Characteristics of cloud computing in the business context: A systematic literature review," *Global J. Flexible Syst. Manag.*, vol. 15, no. 1, pp. 59–68, Mar 2014. [Online]. Available: https://doi.org/10.1007/s40171-013-0055-4

[26] H. F. Atlam, R. J. Walters, and G. B. Wills, "Fog computing and the Internet of Things: A review," *Big Data Cogn. Comput.*, vol. 2, no. 2, p. 10, 2018.

[27] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. 1st Ed. MCC Workshop Mobile Cloud Comput.*, 2012, pp. 13–16.

[28] B. Varghese, N. Wang, D. S. Nikolopoulos, and R. Buyya. "'Feasibility of Fog Computing." 2017. [Online]. Available: http://arxiv.org/abs/1701.05451

[29] M. Iorga, L. Feldman, R. Barton, M. J. Martin, N. S. Goren, and C. Mahmoudi, "Fog computing conceptual model," Nat. Inst. Sci. Technol., Gaithersburg, MD, USA, Rep. 500-325, 2018.

[30] H. Sabireen and V. Neelanarayanan, "A review on fog computing: Architecture, fog with IoT, algorithms and research challenges," *ICT Exp.*, vol. 7, no. 2, pp. 162–176, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405959521000606

[31] Z. Hao, E. Novak, S. Yi, and Q. Li, "Challenges and software architecture for fog computing," *IEEE Internet Comput.*, vol. 21, no. 2, pp. 44–53, Mar./Apr. 2017.

[32] S. Hamdan, M. Ayyash, and S. Almajali, "Edge-computing architectures for Internet of Things applications: A survey," *Sensors*, vol. 20, no. 22, p. 6441, Nov. 2020. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/33187267

[33] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[34] Y. Ai, M. Peng, and K. Zhang, "Edge computing technologies for Internet of Things: A primer," *Digit. Commun. Netw.*, vol. 4, no. 2, pp. 77–86, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352864817301335

[35] "Cisco." Accessed: 2023. [Online]. Available: https://newsroom.cisco.com/press-release_content?articleId=1365576

[36] M. Chiang, S. Ha, F. Risso, T. Zhang, and I. Chih-Lin, "Clarifying fog computing and networking: 10 questions and answers," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 18–20, Apr. 2017.

[37] A. Leivadeas, G. Kesidis, M. Ibnkahla, and I. Lambadaris, "VNF placement optimization at the edge and cloud," *Future Internet*, vol. 11, no. 3, p. 69, 2019. [Online]. Available: https://www.mdpi.com/1999-5903/11/3/69

[38] F. B. Jemaa, G. Pujolle, and M. Pariente, "QoS-aware VNF placement optimization in edge-central carrier cloud architecture," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2016, pp. 1–7.

[39] Q. Zhang, L. Liu, C. Pu, Q. Dou, L. Wu, and W. Zhou, "A comparative study of containers and virtual machines in big data environment," in *Proc. IEEE 11th Int. Conf. Cloud Comput. (CLOUD)*, 2018, pp. 178–185.

[40] "Production-Grade Container Orchestration." 2019. [Online]. Available: https://kubernetes.io/

[41] "Scroll Viewport and Atlassian Confluence." 2022. [Online]. Available: https://docs.d2iq.com/

[42] J. Shen and J. Brower, *Access and Edge Network Architecture and Management*. Cham, Switzerland: Springer Int., 2021, pp. 157–183. [Online]. Available: https://doi.org/10.1007/978-3-030-81961-3_5

[43] S. Ramanathan et al., "Demonstration of containerized central unit live migration in 5G radio access network," in *Proc. IEEE 8th Int. Conf. Netw. Softw. (NetSoft)*, 2022, pp. 225–227.

[44] X. Li et al., "Novel resource and energy management for 5G integrated backhaul/fronthaul (5G-crosshaul)," in *Proc. Int. Workshop 5G RAN Design (ICC)*, 2017, pp. 1–7. [Online]. Available: http://porto.polito.it/2665423/

[45] Z. Usmani and S. Singh, "A survey of virtual machine placement techniques in a cloud data center," *Procedia Comput. Sci.*, vol. 78, pp. 491–498, Jan. 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050916000958

[46] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network slicing in 5G: Survey and challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 94–100, May 2017.

[47] Q. Wang and J. A. Calero, *5G PPP View on 5G Architecture*, Eur. Commission, Brussels, Belgium, 2016.

[48] N. Nikaein et al., "Network store: Exploring slicing in future 5G networks," in *Proc. 10th Int. Workshop Mobility Evolving Internet Architect. (MobiArch)*, 2015, pp. 8–13.

[49] K. Katsalis, N. Nikaein, E. Schiller, R. Favraud, and T. I. Braun, "5G architectural design patterns," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, May 2016, pp. 32–37.

[50] N.-T. Dinh and Y. Kim, "An efficient availability guaranteed deployment scheme for IoT service chains over fog-core cloud networks," *Sensors*, vol. 18, no. 11, p. 3970, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/11/3970

[51] B. Blanco, I. Taboada, J. O. Fajardo, and F. Liberal, "A robust optimization based energy-aware virtual network function placement proposal for small cell 5G networks with mobile edge computing capabilities," *Mobile Inf. Syst.*, vol. 2017, Oct. 2017, Art. no. 2603410.

[52] S. Agarwal, F. Malandrino, C. F. Chiasserini, and S. De, "VNF placement and resource allocation for the support of vertical services in 5G networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 1, pp. 433–446, Feb. 2019.

[53] D. Dwiardhika and T. Tachibana, "Virtual network embedding based on security level with VNF placement," *Security Commun. Netw.*, vol. 2019, Feb. 2019, Art. no. 5640134.

[54] J. Yusupov, A. Ksentini, G. Marchetto, and R. Sisto, "Multi-objective function splitting and placement of network slices in 5G mobile networks," in *Proc. IEEE Conf. Stand. Commun. Netw. (CSCN)*, 2018, pp. 1–6.

[55] S. Agarwal, F. Malandrino, C. Chiasserini, and S. De, "Joint VNF placement and CPU allocation in 5G," in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, 2018, pp. 1943–1951.

[56] L. Yala, P. A. Frangoudis, and A. Ksentini, "Latency and availability driven VNF placement in a MEC-NFV environment," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–7.

[57] T. PARK, "A two-phase heuristic algorithm for determining buffer sizes of production lines," *Int. J. Prod. Res.*, vol. 31, no. 3, pp. 613–631, 1993.

[58] D. Qi, S. Shen, and G. Wang, "Towards an efficient VNF placement in network function virtualization," *Comput. Commun.*, vol. 138, pp. 81–89, Apr. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366418308247

[59] Y. Mu, L. Wang, and J. Zhao, "Energy-efficient and interference-aware VNF placement with deep reinforcement learning," in *Proc. IFIP Netw. Conf. (IFIP Netw.)*, 2021, pp. 1–9.

[60] G. L. Santos, T. Lynn, J. Kelner, and P. T. Endo, "Availability-aware and energy-aware dynamic SFC placement using reinforcement learning," *J. Supercomput.*, vol. 77, no. 11, pp. 12711–12740, Nov. 2021. [Online]. Available: https://doi.org/10.1007/s11227-021-03784-7

[61] R. Solozabal, J. Ceberio, A. Sanchoyerto, L. Zabala, B. Blanco, and F. Liberal, "Virtual network function placement optimization with deep reinforcement learning," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 2, pp. 292–303, Feb. 2020.

[62] C. Pham, N. H. Tran, S. Ren, W. Saad, and C. S. Hong, "Traffic-aware and energy-efficient VNF placement for service chaining: Joint sampling and matching approach," *IEEE Trans. Services Comput.*, vol. 13, no. 1, pp. 172–185, Jan./Feb. 2020.

[63] M. Mechtri, C. Ghribi, and D. Zeghlache, "VNF placement and chaining in distributed cloud," in *Proc. IEEE 9th Int. Conf. Cloud Comput. (CLOUD)*, Jun. 2016, pp. 376–383.

[64] 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 Commun. Mag.*, vol. 53, no. 12, pp. 60–66, Dec. 2015.

[65] V. Sciancalepore, F. Z. Yousaf, and X. Costa-Perez, "z-TORCH: An automated NFV orchestration and monitoring solution," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 4, pp. 1292–1306, Dec. 2018.

[66] Y. Yao, S. Guo, P. Li, G. Liu, and Y. Zeng, "Forecasting assisted VNF scaling in NFV-enabled networks," *Comput. Netw.*, vol. 168, Feb. 2020, Art. no. 107040. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128619305705

[67] B. Nemeth, N. Molner, J. Martinperez, C. J. Bernardos, A. De la Oliva, and B. Sonkoly, "Delay and reliability-constrained VNF placement on mobile and volatile 5G infrastructure," *IEEE Trans. Mobile Comput.*, vol. 21, no. 9, pp. 3150–3162, Sep. 2022.

[68] O. Alhussein et al., "Joint VNF placement and multicast traffic routing in 5G core networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2018, pp. 1–6.

[69] N. Kiran, X. Liu, S. Wang, and C. Yin, "VNF placement and resource allocation in SDN/NFV-enabled MEC networks," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, 2020, pp. 1–6.

[70] M. A. Abdelaal, G. A. Ebrahim, and W. R. Anis, "Efficient placement of service function chains in cloud computing environments," *Electronics*, vol. 10, no. 3, p. 323, 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/3/323

[71] O. A. Wahab, N. Kara, C. Edstrom, and Y. Lemieux, "MAPLE: A machine learning approach for efficient placement and adjustment of virtual network functions," *J. Netw. Comput. Appl.*, vol. 142, pp. 37–50, Sep. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804519301924

[72] G. Kibalya, J. Serrat, J.-L. Gorricho, D. G. Bujjingo, J. Sserugunda, and P. Zhang, "A reinforcement learning approach for placement of stateful virtualized network functions," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manag. (IM)*, 2021, pp. 672–676.

[73] G. Yuan et al., "Fault tolerant placement of stateful VNFs and dynamic fault recovery in cloud networks," *Comput. Netw.*, vol. 166, Jan. 2020, Art. no. 106953. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128619300970

[74] T. Kuo, B. Liou, K. C. Lin, and M. Tsai, "Deploying chains of virtual network functions: On the relation between link and server usage," in *Proc. IEEE INFOCOM 35th Annu. IEEE Int. Conf. Comput. Commun.*, 2016, pp. 1–9.

[75] A. Mohammadkhan, S. Ghapani, G. Liu, W. Zhang, K. K. Ramakrishnan, and T. Wood, "Virtual function placement and traffic steering in flexible and dynamic software defined networks," in *Proc. 21st IEEE Int. Workshop Local Metropolitan Area Netw.*, Apr. 2015, pp. 1–6.

[76] M. Mechtri, C. Ghribi, and D. Zeghlache, "A scalable algorithm for the placement of service function chains," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 3, pp. 533–546, Sep. 2016.

[77] M. Leconte, G. S. Paschos, P. Mertikopoulos, and U. C. Kozat, "A resource allocation framework for network slicing," in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, 2018, pp. 2177–2185.

[78] J. Cao, Y. Zhang, W. An, X. Chen, J. Sun, and Y. Han, "VNF-FG design and VNF placement for 5G mobile networks," *Sci. China Inf. Sci.*, vol. 60, no. 4, 2017, Art. no. 040302.

[79] 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. 13th Int. Conf. Netw. Service Manag. (CNSM)*, 2017, pp. 1–7.

[80] S. I. Kim and H. S. Kim, "A VNF placement method based on VNF characteristics," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, 2021, pp. 864–869.

[81] R. Behravesh, D. Harutyunyan, E. Coronado, and R. Riggio, "Time-sensitive mobile user association and SFC placement in MEC-enabled 5G networks," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 3, pp. 3006–3020, Sep. 2021.

[82] P. P. Ray and N. Kumar, "SDN/NFV architectures for edge-cloud oriented IoT: A systematic review," *Comput. Commun.*, vol. 196, pp. 129–153, Mar. 2021.

[83] S. M. A. Araújo, F. S. H. de Souza, and G. R. Mateus, "A hybrid optimization-machine learning approach for the VNF placement and chaining problem," *Comput. Netw.*, vol. 199, Nov. 2021, Art. no. 108474. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128621004229

[84] H. Halabian, "Distributed resource allocation optimization in 5G virtualized networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 627–642, Mar. 2019.

[85] Y. L. Lee, J. Loo, T. C. Chuah, and L.-C. Wang, "Dynamic network slicing for multitenant heterogeneous cloud radio access networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2146–2161, Apr. 2018.

[86] R. Su et al., "Resource allocation for network slicing in 5G telecommunication networks: A survey of principles and models," *IEEE Netw.*, vol. 33, no. 6, pp. 172–179, Nov./Dec. 2019.

[87] A. De Domenico, Y.-F. Liu, and W. Yu, "Optimal virtual network function deployment for 5G network slicing in a hybrid cloud infrastructure," *IEEE Trans. Wireless Commun.*, vol. 19, no. 12, pp. 7942–7956, Dec. 2020.

[88] S. T. Arzo, R. Bassoli, F. Granelli, and F. H. P. Fitzek, "Study of virtual network function placement in 5G cloud radio access network," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 4, pp. 2242–2259, Dec. s2020.

[89] S. S. Shinde, D. Marabissi, and D. Tarchi, "A network operator-biased approach for multi-service network function placement in a 5G network slicing architecture," *Comput. Netw.*, vol. 201, Dec. 2021, Art. no. 108598. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128621004989

[90] A. Dalgkitsis, P.-V. Mekikis, A. Antonopoulos, G. Kormentzas, and C. Verikoukis, "Dynamic resource aware VNF placement with deep reinforcement learning for 5G networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2020, pp. 1–6.

[91] L. Gu, D. Zeng, W. Li, S. Guo, A. Zomaya, and H. Jin, "Deep reinforcement learning based VNF management in geo-distributed edge computing," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2019, pp. 934–943.

[92] P. T. A. Quang, Y. Hadjadj-Aoul, and A. Outtagarts, "A deep reinforcement learning approach for VNF forwarding graph embedding," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 4, pp. 1318–1331, Dec. 2019.

[93] P. T. A. Quang, A. Bradai, K. D. Singh, and Y. Hadjadj-Aoul, "Multi-domain non-cooperative VNF-FG embedding: A deep reinforcement learning approach," in *Proc. IEEE INFOCOM IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2019, pp. 886–891.

[94] S. I. Kim and H. S. Kim, "A research on dynamic service function chaining based on reinforcement learning using resource usage," in *Proc. 9th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, 2017, pp. 582–586.

[95] O. Houidi, O. Soualah, W. Louati, and D. Zeghlache, "An enhanced reinforcement learning approach for dynamic placement of virtual network functions," in *Proc. IEEE 31st Annu. Int. Symp. Pers. Indoor Mobile Radio Commun.*, 2020, pp. 1–7.

[96] Q. Zhang, F. Liu, and C. Zeng, "Adaptive interference-aware VNF placement for service-customized 5G network slices," in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, Apr. 2019, pp. 2449–2457.

[97] M. Golkarifard, C. F. Chiasserini, F. Malandrino, and A. Movaghar, "Dynamic VNF placement, resource allocation and traffic routing in 5G," *Comput. Netw.*, vol. 188, Apr. 2021, Art. no. 107830. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128621000177

[98] Z. Xiang, F. Gabriel, E. Urbano, G. T. Nguyen, M. Reisslein, and F. H. P. Fitzek, "Reducing latency in virtual machines: Enabling tactile Internet for human–machine co-working," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1098–1116, May 2019.

[99] I. Sanchez-Navarro, J. M. A. Calero, and Q. Wang, "Topology awareness for smart 5G eMBB network slicing VNF placement," in *Proc. IEEE 21st Int. Symp. World Wireless Mobile Multimedia Netw. (WoWMoM)*, 2020, pp. 415–420.

[100] J. J. A. Esteves, A. Boubendir, F. Guillemin, and P. Sens, "Location-based data model for optimized network slice placement," in *Proc. 6th IEEE Conf. Netw. Softw. (NetSoft)*, 2020, pp. 404–412.

[101] Z. Yan, J. Ge, Y. Wu, L. Li, and T. Li, "Automatic virtual network embedding: A deep reinforcement learning approach with graph convolutional networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 6, pp. 1040–1057, Jun. 2020.

[102] J. J. A. Esteves, A. Boubendir, F. Guillemin, and P. Sens, "DRL-based slice placement under non-stationary conditions," in *Proc. IEEE 17th Int. Conf. Netw. Service Manag. (CNSM)*, 2021, pp. 225–233.

[103] Z. Zhang, G. Wu, and H. Ren, "Multi-attribute-based QoS-aware virtual network function placement and service chaining algorithms in smart cities," *Comput. Elect. Eng.*, vol. 96, Dec. 2021, Art. no. 107465. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790621004237

[104] Q. Xu, D. Gao, T. Li, and H. Zhang, "Low latency security function chain embedding across multiple domains," *IEEE Access*, vol. 6, pp. 14474–14484, 2018.

[105] P. M. Mohan and M. Gurusamy, "Resilient VNF placement for service chain embedding in diversified 5G network slices," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2019, pp. 1–6.

[106] A. Hmaity, M. Savi, F. Musumeci, M. Tornatore, and A. Pattavina, "Virtual network function placement for resilient service chain provisioning," in *Proc. 8th Int. Workshop Resilient Netw. Design Model. (RNDM)*, Sep. 2016, pp. 245–252.

[107] S. Imadali and A. Bousselmi, "Cloud native 5G virtual network functions: Design principles and use cases," in *Proc. IEEE 8th Int. Symp. Cloud Service Comput. (SC2)*, 2018, pp. 91–96.

[108] S. Rizou et al., "Programmable edge-to-cloud virtualization for 5G media industry: The 5G-MEDIA approach," in *Proc. Artif. Intell. Appl. Innov. (AIAI) IFIP WG Int. Workshops*, 2020, pp. 95–104.

[109] S. Aleyadeh, A. Moubayed, P. Heidari, and A. Shami, "Optimal container migration/re-instantiation in hybrid computing environments," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 15–30, 2022.

[110] A. Singh, G. S. Aujla, and R. S. Bali, "Container-based load balancing for energy efficiency in software-defined edge computing environment," *Sustain. Comput. Inf. Syst.*, vol. 30, Jun. 2021, Art. no. 100463. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210537920301876

[111] M. Xu, W. Tian, and R. Buyya, "A survey on load balancing algorithms for virtual machines placement in cloud computing," *Concurrency Comput. Pract. Exp.*, vol. 29, no. 12, 2017, Art. no. e4123.

[112] Z. Wu, Y. Deng, H. Feng, Y. Zhou, and G. Min, "Blender: A traffic-aware container placement for containerized data centers," in *Proc. Design Autom. Test Europe Conf. Exhibit. (DATE)*, 2021, pp. 986–989.

[113] J. Darrous, T. Lambert, and S. Ibrahim, "On the importance of container image placement for service provisioning in the edge," in *Proc. 28th Int. Conf. Comput. Commun. Netw. (ICCCN)*, 2019, pp. 1–9.

[114] Y. Xiao and M. Krunz, "Dynamic network slicing for scalable fog computing systems with energy harvesting," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 12, pp. 2640–2654, Dec. 2018.

[115] Y. Zhai, T. Bao, L. Zhu, M. Shen, X. Du, and M. Guizani, "Toward reinforcement-learning-based service deployment of 5G mobile edge computing with request-aware scheduling," *IEEE Wireless Commun.*, vol. 27, no. 1, pp. 84–91, Feb. 2020.

[116] A. Mseddi, W. Jaafar, H. Elbiaze, and W. Ajib, "Joint container placement and task provisioning in dynamic fog computing," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10028–10040, Dec. 2019.

[117] P.-J. Maenhaut, B. Volckaert, V. Ongenae, and F. D. Turck, "Resource management in a containerized cloud: Status and challenges," *J. Netw. Syst. Manag.*, vol. 28, pp. 197–246, Nov. 2019.

[118] L. Lv et al., "Communication-aware container placement and reassignment in large-scale Internet data centers," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 540–555, Mar. 2019.

[119] M.-K. Hussein, M.-H. Mousa, and M. A. Alqarni, "A placement architecture for a container as a service (CaaS) in a cloud environment," *J. Cloud Comput.*, vol. 8, pp. 1–15, May 2019.

[120] A. Pires, J. Simão, and L. Veiga, "Distributed and decentralized orchestration of containers on edge clouds," *J. Grid Comput.*, vol. 19, no. 3, p. 36, Jul. 2021. [Online]. Available: https://doi.org/10.1007/s10723-021-09575-x

[121] M. Selimi. L. Cerdá-Alabern, M. Sánchez-Artigas, F. Freitag, and L. Veiga, "Practical service placement approach for microservices architecture," in *Proc. 17th IEEE/ACM Int. Symp. Clust. Cloud Grid Comput. (CCGRID)*, 2017, pp. 401–410.

[122] S. Arora and A. Ksentini, "Dynamic resource allocation and placement of cloud native network services," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2021, pp. 1–6.

[123] Y. Alahmad, A. Agarwal, M. Zaman, and N. Goel, "Container placement for resource utilization in cyber physical cloud systems," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, 2021, pp. 1–6.

[124] M. Abdullah, W. Iqbal, F. Bukhari, and A. Erradi, "Diminishing returns and deep learning for adaptive CPU resource allocation of containers," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 4, pp. 2052–2063, Dec. 2020.

[125] R. C. Chiang, "Contention-aware container placement strategy for docker swarm with machine learning based clustering algorithms," *Clust. Comput.*, vol. 26, pp. 13–23, Feb. 2023.

[126] R. Zhang, Y. Chen, B. Dong, F. Tian, and Q. Zheng, "A genetic algorithm-based energy-efficient container placement strategy in CaaS," *IEEE Access*, vol. 7, pp. 121360–121373, 2019.

[127] A. Havet, V. Schiavoni, P. Felber, M. Colmant, R. Rouvoy, and C. Fetzer, "GenPack: A generational scheduler for cloud data centers," in *Proc. IEEE Int. Conf. Cloud Eng. (IC2E)*, 2017, pp. 95–104.

[128] S. Tuli, S. R. Poojara, S. N. Srirama, G. Casale, and N. R. Jennings, "COSCO: Container orchestration using co-simulation and gradient based optimization for fog computing environments," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 1, pp. 101–116, Jan. 2022.

[129] A. Al-Moalmi, J. Luo, A. Salah, K. Li, and L. Yin, "A whale optimization system for energy-efficient container placement in data centers," *Exp. Syst. Appl.*, vol. 164, Feb. 2021, Art. no. 113719. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417420305431

[130] R. Zhang, Y. Chen, F. Zhang, F. Tian, and B. Dong, "Be good neighbors: A novel application isolation metric used to optimize the initial container placement in CaaS," *IEEE Access*, vol. 8, pp. 178195–178207, 2020.

[131] T. Menouer and P. Darmon, "New scheduling strategy based on multi-criteria decision algorithm," in *Proc. 27th Euromicro Int. Conf. Parallel Distrib. Netw. Process. (PDP)*, 2019, pp. 101–107.

[132] S. B. Nath, S. K. Addya, S. Chakraborty, and S. K. Ghosh, "Green containerized service consolidation in cloud," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2020, pp. 1–6.

[133] B. Tan, H. Ma, Y. Mei, and M. Zhang, "A cooperative coevolution genetic programming hyper-heuristic approach for on-line resource allocation in container-based clouds," *IEEE Trans. Cloud Comput.*, vol. 10, no. 3, pp. 1500–1514, Jul./Sep. 2022.

[134] R. Zhou, Z. Li, and C. Wu, "An efficient online placement scheme for cloud container clusters," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1046–1058, May 2019.

[135] F. Neves, R. Vilaça, and J. Pereira, "Black-box inter-application traffic monitoring for adaptive container placement," in *Proc. 35th Annu. ACM Symp. Appl. Comput. (SAC)*, 2020, pp. 259–266. [Online]. Available: https://doi.org/10.1145/3341105.3374007

[136] Y. Wu and H. Chen, "ABP scheduler: Speeding up service spread in docker swarm," in *Proc. IEEE Int. Symp. Parallel Distrib. Process. Appl. IEEE Int. Conf. Ubiquitous Comput. Commun. (ISPA/IUCC)*, 2017, pp. 691–698.

[137] L. Johansson, "Latency and traffic aware container placement in distributed cloud," Dept. Comput. Sci., Linköping Univ., Linköping, Sweden, 2019.

[138] A. M. Maliszewski, A. Vogel, D. Griebler, E. Roloff, L. G. Fernandes, and O. A. N. Philippe, "Minimizing communication overheads in container-based clouds for HPC applications," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, 2019, pp. 1–6.

[139] Y. Mao, J. Oak, A. Pompili, D. Beer, T. Han, and P. Hu, "DRAPS: Dynamic and resource-aware placement scheme for docker containers in a heterogeneous cluster," in *Proc. IEEE 36th Int. Perform. Comput. Commun. Conf. (IPCCC)*, 2017, pp. 1–8.

[140] I. Lera, C. Guerrero, and C. Juiz, "Availability-aware service placement policy in fog computing based on graph partitions," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3641–3651, Apr. 2019.

[141] E. H. Bourhim, H. Elbiaze, and M. Dieye, "Inter-container communication aware container placement in fog computing," in *Proc. 15th Int. Conf. Netw. Service Manag. (CNSM)*, 2019, pp. 1–6.

[142] M. Gill and D. Singh, "Aco based container placement for CaaS in fog computing," *Procedia Comput. Sci.*, vol. 167, pp. 760–768, Jan. 2020.

[143] A. M. Hafez, A. Abdelsamea, A. A. El-Moursy, S. M. Nassar, and M. B. E. Fayek, "Modified ant colony placement algorithm for containers," in *Proc. 15th Int. Conf. Comput. Eng. Syst. (ICCES)*, 2020, pp. 1–6.

[144] S. Sobhanayak, K. Jaiswal, A. K. Turuk, B. Sahoo, B. K. Mohanta, and D. Jena, "Container-based task scheduling for edge computing in IoT-cloud environment using improved HBF optimisation algorithm," *Int. J. Embedded Syst.*, vol. 13, no. 1, pp. 85–100, 2020.

[145] T. Rausch, A. Rashed, and S. Dustdar, "Optimized container scheduling for data-intensive serverless edge computing," *Future Gener. Comput. Syst.*, vol. 114, pp. 259–271, Jan. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X2030399X

[146] D. Harutyunyan, N. Shahriar, R. Boutaba, and R. Riggio, "Latency-aware service function chain placement in 5G mobile networks," in *Proc. IEEE Conf. Netw. Softw. (NetSoft)*, 2019, pp. 133–141.

[147] M. A. Abdelaal, G. A. Ebrahim, and W. R. Anis, "Efficient placement of service function chains in cloud computing environments," *Electronics*, vol. 10, no. 3, p. 323, 2021.

[148] J. Santos, T. Wauters, B. Volckaert, and F. De Turck, "Towards delay-aware container-based service function chaining in fog computing," in *Proc. IEEE/IFIP Netw. Oper. Manag. Symp. (NOMS)*, 2020, pp. 1–9.

[149] N. Kazemifard and V. Shah-Mansouri, "Minimum delay function placement and resource allocation for open RAn (O-RAN) 5G networks," *Comput. Netw.*, vol. 188, Apr. 2021, Art. no. 107809. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128621000037

[150] M. Goudarzi, M. S. Palaniswami, and R. Buyya, "A distributed deep reinforcement learning technique for application placement in edge and fog computing environments," *IEEE Trans. Mobile Comput.*, early access, Oct. 27, 2021, doi: 10.1109/TMC.2021.3123165.

[151] J. Kim, J. Lee, T. Kim, and S. Pack, "Deep reinforcement learning based cloud-native network function placement in private 5G in etworks," in *Proc. IEEE Globecom Workshops (GC Wkshps*, 2020, pp. 1–6.

[152] P. Kochovski, R. Sakellariou, M. Bajec, P. Drobintsev, and V. Stankovski, "An architecture and stochastic method for database container placement in the edge-fog-cloud continuum," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. (IPDPS)*, 2019, pp. 396–405.

[153] M. Haghi Kashani, A. M. Rahmani, and N. Jafari Navimipour, "Quality of service-aware approaches in fog computing," *Int. J. Commun. Syst.*, vol. 33, no. 8, 2020, Art. no. e4340.

[154] S. Bhaumik and S. Chakraborty, "Managing container QoS with network and storage workloads over a hyperconverged platform," in *Proc. IEEE 45th Conf. Local Comput. Netw. (LCN)*, 2020, pp. 112–123.

[155] A. C. Caminero and R. Muñoz-Mansilla, "Quality of service provision in fog computing: Network-aware scheduling of containers," *Sensors*, vol. 21, no. 12, p. 3978, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/12/3978

[156] S. Shringarputale, P. McDaniel, K. Butler, and T. La Porta, "Co-residency attacks on containers are real," in *Proc. ACM SIGSAC Conf. Cloud Comput. Security Workshop (CCSW)*, 2020, pp. 53–66. [Online]. Available: https://doi.org/10.1145/3411495.3421357

[157] T. Kong, L. Wang, D. Ma, Z. Xu, Q. Yang, and K. Chen, "A secure container deployment strategy by genetic algorithm to defend against co-resident attacks in cloud computing," in *Proc. IEEE 21st Int. Conf. High Perform. Comput. Commun. IEEE 17th Int. Conf. Smart City IEEE 5th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, 2019, pp. 1825–1832.

[158] R. Urgaonkar, S. Wang, T. He, M. Zafer, K. Chan, and K. K. Leung, "Dynamic service migration and workload scheduling in edge-clouds," *Perform. Eval.*, vol. 91, pp. 205–228, Sep. 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166531615000619

[159] J. Luo et al., "Container-based fog computing architecture and energy-balancing scheduling algorithm for energy IoT," *Future Gener. Comput. Syst.*, vol. 97, pp. 50–60, Aug. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X1831358X

[160] M. Ramasamy, M. Balakrishnan, and C. Thangaraj, "Priority queue scheduling approach for resource allocation in containerized clouds," in *Inventive Computation Technologies*, S. Smys, R. Bestak, and Á. Rocha, Eds. Cham, Switzerland: Springer Int., 2020, pp. 758–765.

[161] Z. Cai and R. Buyya, "Inverse queuing model based feedback control for elastic container provisioning of Web systems in kubernetes," *IEEE Trans. Comput.*, vol. 71, no. 2, pp. 337–348, Feb. 2022.

[162] L. Toka, G. Dobreff, B. Fodor, and B. Sonkoly, "Machine learning-based scaling management for kubernetes edge clusters," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 1, pp. 958–972, Mar. 2021.

[163] J.-E. Dartois, J. Boukhobza, A. Knefati, and O. Barais, "Investigating machine learning algorithms for modeling SSD I/O performance for container-based virtualization," *IEEE Trans. Cloud Comput.*, vol. 9, no. 3, pp. 1103–1116, Jul./Sep. 2021.

[164] Y. Hu, H. Zhou, C. de Laat, and Z. Zhao, "Concurrent container scheduling on heterogeneous clusters with multi-resource constraints," *Future Gener. Comput. Syst.*, vol. 102, pp. 562–573, Jan. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X19309446

[165] A. Asensio, X. Masip-Bruin, J. Garcia, and S. Sánchez, "On the optimality of concurrent container clusters scheduling over heterogeneous smart environments," *Future Gener. Comput. Syst.*, vol. 118, pp. 157–169, May 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X21000054

Wissal Attaoui (Member, IEEE) received the engineering degree in telecommunications and network engineering from the National School of Applied Sciences of Tangier, Morocco, in 2015, and the Ph.D. degree from the National Higher School of Electricity and Mechanics (Ensem) in 2022. She is an Intelligent Network Operations Manager with INWI Moroccan Telecom Operator. She has published many conference and journal papers. Her research interests include mobile cloud computing, virtual machine placement, VNF placement in 5G, initial access, and beam alignment in mmWave/terahertz 5G/6G wireless communication networks. She received the Best Paper Award at IEEE IWCMC'19. She served as a Reviewer for many international conferences and journals, such as IEEE ICC'21, IEEE Globecom, and IEEE ACCESS.

Essaid Sabir (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees (Hons.) in ECE from the Mohammed V University of Rabat, Morocco, in 2004 and 2007, respectively, and the Ph.D. degree (Hons.) in networking and computer engineering from Avignon University, France, in 2010. He was a non-tenure-track Assistant Professor with Avignon University from 2009 to 2012, and was a Professor with ENSEM, Hassan II University of Casablanca, where he was leading the NEST Research Group from 2012 to 2022. He is currently with the Department of Computer Science, University of Quebec at Montreal. He is/was the Main Investigator of many research projects, and has been involved in several other national and international projects. His research interests include wireless networks and mobile communications, 5G and beyond, IoT, infrastructure-less networking, ultrareliable low-latency communications, wireless artificial intelligence, and networking games. His work has been awarded in many international venues, such as IEEE 5GWF'21, IEEE IWCMC'19, IEEE and WF-IoT'20. As an attempt to bridge the gap between academia and industry, he founded the International Conference on Ubiquitous Networking and co-founded the International Conference on Wireless Networks and Mobile Communications. He serves as an associate editor and/or reviewer for many international journals. He organized numerous events and served as the general chair, TPC chair, and other executive roles for many major international events.

Halima Elbiaze (Senior Member, IEEE) received the M.Sc. degree in telecommunication systems from the Université de Versailles in 1998, and the Ph.D. degree in computer science from Telecom Sud Paris, France, in 2002. She is currently a Full Professor with the Department of Computer Science, Université du Québec à Montréal, where she has been serving since 2003. She has authored and coauthored many journals, conference papers, and patents. Her research interests include network performance evaluation, 5G and beyond, IoT, traffic engineering, cloud/edge computing, and quality-of-service management in optical and wireless networks.

Mohsen Guizani (Fellow, IEEE) received the B.S. (with Distinction), M.S., and Ph.D. degrees in electrical and computer engineering from Syracuse University, Syracuse, NY, USA, in 1985, 1987, and 1990, respectively. He is currently a Professor of Machine Learning with the Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE. Previously, he worked in different institutions in the USA. He is the author of 11 books, more than 1000 publications and several U.S. patents. His research interests include applied machine learning and artificial intelligence, smart city, Internet of Things, intelligent autonomous systems, and cybersecurity. He was listed as a Clarivate Analytics Highly Cited Researcher in Computer Science in 2019–2022. He has won several research awards, including the 2015 IEEE Communications Society Best Survey Paper Award, the Best ComSoc Journal Paper Award in 2021, as well as five Best Paper Awards from ICC and Globecom Conferences. He is also the recipient of the 2017 IEEE Communications Society Wireless Technical Committee Recognition Award, the 2018 AdHoc Technical Committee Recognition Award, and the 2019 IEEE Communications and Information Security Technical Recognition Award. He served as the Editor-in-Chief for IEEE NETWORK and is currently serving on the editorial boards of many IEEE TRANSACTIONS and magazines. He was the Chair of the IEEE Communications Society Wireless Technical Committee and the TAOS Technical Committee. He served as the IEEE Computer Society Distinguished Speaker and is currently the IEEE ComSoc Distinguished Lecturer.