On Green Edge Computing with Machine Learning Applications

by

Francesco Spinelli

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in

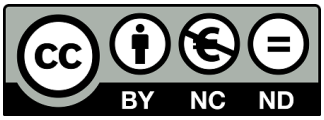
Telematic Engineering

Universidad Carlos III de Madrid

Advisor/Tutor:

Vincenzo Mancuso

April 2024



ACKNOWLEDGEMENTS

First of all, I would like to express my deepest gratitude to my advisor Prof. Vincenzo Mancuso for giving me the possibility to pursue my PhD at IMDEA Networks. His invaluable guidance, expertise, patience, and insightful feedback have been instrumental in shaping the direction of my PhD journey and my future research career. I would like also to thank him for giving me the possibility to do an internship abroad, which resulted in an unforgettable life experience. On the same line, I would like to thank Prof. Katia Obraczka for hosting me at the University of California Santa Cruz and for allowing me to see how research is accomplished in another part of the world. Furthermore, I want to thank also all the members of the i-NRG lab for making me feel at home (Hari, Solange, Lakshmi, Andrea, and Dylan) and for all the good laughs and trips we had. Hope to see you soon dudes!

Towards half of my PhD, I also have the pleasure to start working with a wonderful post-doc, Dr. Antonio Bazco-Nogueras. I am grateful to him for all his assistance and guidance and also for his extreme patience towards all my doubts and errors. Thank you! I want to thank all the friends and colleagues who supported me during these years, from the people on the 1st floor at IMDEA to the Italian community that made these years unforgettable and full of good experiences (in no particular order: Dario, Alessia, Maurizio, Gaia, Gaetano, Christian, Pippo, Giulia, Losky, Vittorio "il comandante", el Rojo, Andrea, Valerio, Boris, Mer, Luigi, Stefano, Federico, Alessio, Federica, Antonio, Mauro, Laura).

I want to deeply thank my parents Giuseppe and Sarah for supporting me throughout all these years with my "crazy" ideas, from going to study in Torino, to living in Paris and in the end to move in Madrid. I want especially to thank my sister Marta for being there every time I needed support during hard times in my PhD Journey. Finally, I want to thank my girlfriend Camila, who made every day shine with her contagious laughter and had a lot of patience and support throughout this experience.

PUBLISHED AND SUBMITTED CONTENT

The ideas and investigations of this thesis resulted in the following refereed publications:

1. **F. Spinelli** and V. Mancuso, "Toward Enabled Industrial Verticals in 5G: A Survey on MEC-Based Approaches to Provisioning and Flexibility," in IEEE Communications Surveys & Tutorials, vol. 23, no. 1, pp. 596-630, Firstquarter 2021, doi: 10.1109/COMST.2020.3037674[1].

   - This work is fully included and its content is reported in Chapter 2.
   - The author fully participated in the writing of the paper and his role in this work focused on performing the literature review and investigation into open problems of MEC and evaluating the proposed smart metropolitan scenario.
   - The material from this source included in this thesis is not singled out with typographic means and references.

2. **F. Spinelli** and V. Mancuso, "A Migration Path Toward Green Edge Gaming," 2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), Belfast, United Kingdom, 2022, pp. 347-356, doi: 10.1109/WoWMoM54355.2022.00033[2].

   - This work is partially included and its content is reported in Chapter 3.
   - The author fully participated in the writing of the paper and his role in this work focused on proposing the scenario, formulating the problem and the design and experimentation of a heuristic.
   - The material from this source included in this thesis is not singled out with typographic means and references.

3. **F. Spinelli**, Antonio Bazco-Nogueras, and Vincenzo Mancuso. "Edge Gaming: A Greening Perspective." Computer Communications 192 (2022): 89-105[3].

   - This work is fully included and its content is reported in Chapter 3.
   - The author participated in writing several parts of the paper and his role in this work focused on the design and experimentation of the updated heuristic in novel scenarios.
   - The material from this source included in this thesis is not singled out with typographic means and references.

4. **Francesco Spinelli**, Antonio Bazco Nogueras, and Vincenzo Mancuso. 2023. Offloading Augmented Reality Tasks with Smart Energy Source-Aware Algorithms at the Edge. In Proceedings of the Int'l ACM Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM '23). Association for Computing Machinery, New York, NY, USA, 73–82. https://doi.org/10.1145/3616388.3617523[4].

    - This work is fully included and its content is reported in Chapter 4.
    - The author participated in writing several parts of the paper. His role in this work focused on proposing the scenario, formulating the problem, and designing, implementing, and experimenting with a deep reinforcement learning-based solution and a heuristic.
    - The material from this source included in this thesis is not singled out with typographic means and references.

5. **F. Spinelli**, L. Iannone and J. Tollet, "Multi-Cloud Chaining with Segment Routing," 2020 IFIP Networking Conference (Networking), Paris, France, 2020, pp. 514-518 [5].

    - This work is not included in the thesis.
    - The author fully participated in the writing of the paper and his role in this work focused on implementing a software router in public clouds and then performing an extensive performance evaluation campaign of a network protocol.

Abstract

Edge computing has been a topic of interest for several decades, with different research directions tackled (e.g., CDNs, cloudlets, fog nodes). However, only with the development of 5G networks, edge computing started to be seamlessly integrated into cellular network ecosystems, thanks also to the efforts made by the European Telecommunications Standard Institute (ETSI) with the standardization of the Multi-access Edge Computing (MEC) paradigm. The presence of computing resources at the edge of the cellular infrastructure will help support the growth of novel use cases in everyday life. Some examples are connected cars, and IoT devices but also novel and computing-hungry use cases such as Augmented Reality (AR), Virtual Reality (VR), cloud Gaming, smart manufacturing, and eHealthcare. However, each of these use cases, or industrial *verticals*, poses significant but also distinct challenges (e.g., low latency guarantees, reliability, and privacy concerns among others). Furthermore, the deployment of edge computing resources in a cellular network comes with several challenges for the network operator. First, the deployment and maintenance of these edge resources are costly, in terms of both Capital Expenditure (CAPEX) and Operating Expenses (OPEX). Therefore network operators should consider novel strategies to grant the edge architecture's monetary sustainability. Next, network operators should deploy novel infrastructures that are also sustainable in terms of carbon footprint. In other terms, stakeholders' infrastructures should decrease their overall carbon emission (i.e., using less energy generated by *brown* sources such as carbon). Different worldwide organizations (United Nations, European Union) are pushing towards this sustainable path, which is considered also one of the key pillars of future (i.e., 6G) cellular networks. A possible solution for network operators could be leveraging renewable sources to generate local *green* energy to power the edge infrastructure. Indeed intermittent renewable sources such as solar/wind could generate virtual energy with no cost for network operators therefore allowing them to jointly tackle the goal of decreasing overall monetary costs and carbon footprint. However, those two goals are conflicting, since a network operator would like to maximize the use of its edge infrastructure by admitting end-users tasks as much as possible for revenue maximization but with the downside of possibly using costly brown energy. One key strategy that could help decrease costs and carbon footprint is the smart allocation of computing resources. Indeed, computing resources at the edge are notorious for being scarce and less powerful than their cloud counterparts, and therefore a smart allocation of these resources is pivotal to sustaining novel computing-hungry use cases while decreasing overall costs for network operators. Some of the novel use cases require lots of computing resources (such as cloud gaming and AR) and therefore the smart allocation of offloaded computing tasks should be carefully evaluated. The rise of paradigms such as Network Function

Virtualization (NFV) allows the softwarization of network functions and applications (using for instance virtual machines or containers), allowing novel techniques and scenarios to emerge such as the migration of computing resources to support novel verticals and strategies.

In this thesis, we answer some of the questions and scenarios highlighted above. In particular, we first give an updated review of the state-of-the-art, focusing on several important aspects of the MEC provisioning (such as standardization efforts, techniques to efficiently deploy and support migration of end-user applications, if and allow the offloading of computing tasks). This also gives us the motivations behind this thesis' goals and contributions. Next, we focus on proposing a novel scenario, green edge gaming, where edge computing resources are partially or completely dependent on renewable sources and they have to accommodate heavy computing tasks coming from gaming devices. Another novelty of this scenario is that, since edge servers are located closely, it is possible the migrate allocated gaming jobs between edge servers, according for instance to the availability of green energy. Next, we leverage powerful machine learning techniques such as Deep Reinforcement Learning (DRL) to propose a DRL-based solution for the allocation and migration of AR tasks at the edge. Since the goal of maximizing the admittance of AR tasks while leveraging as much as possible the green energy availability is conflicting, we use a proportional fairness structure, which, thanks to the DRL approach, helps to find a sweet spot between these two goals compared to greedy heuristics. In conclusion, in this thesis, we propose two novel solutions to tackle the problem of allocation and migration jobs in an edge infrastructure, where edge servers depend on intermittent renewable sources. Since one of the key pillars of 6G networks is sustainability, this thesis could lay the foundation for more studies in this evolving scenario.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ADAS Advanced Driver-Assistance System

AF Application Function

AoI Age of Information

AP Access Point

AR/VR Augmented Reality/Virtual Reality

CAM Cooperative Awareness Messages

CAPEX Capital expenditure

CDN Content Delivery Network

CFS Customer Facing Service

CPS Cyber Physical Systems

C-V2X Cellular Vehicle to Everything

DENM Decentralized Environmental Notification Message

DL Downlink

DRL Deep Reinforcement Learning

DNN Deep Neural Network

DT Digital Twin

D2D Device to Device

EH Energy Harvesting

ETSI European Telecommunications Standards Institute

FL Federated Learning

FMeC Follow Me edge Cloud

GNN Graph Neural Network

LADN Local Area Data Network

LEO Low Earth Orbit

LISP Locator/Identifier Separation Protocol

ICN Information Centric Network

IIoT Industrial Internet of Things

ILP Integer Linear Programming

IoT Internet of Things

ISG Industry Specification Group

ITS Intelligent Transport Systems

MANO NFV Management and Network Orchestration

MAR Mobile Augmented Reality

MDP Markov Decision Process

MEAO Mobile Edge Application Orchestrator

MEC Multi-access Edge Computing

MEO Multi-access Edge Orchestrator

MEPM-V MEC Platform-NFV

ML Machine Learning

mmWave Millimeter Wave

M2M Machine to Machine

NEF Network Exposure Function

NFV Network Function Virtualization

NFVI NFV Infrastructure

NFVO NFV Orchestrator

NOMA Non-Orthogonal Multiple Access

OPEX Operating expense

OSS Operations Support System

QoE Quality of Experience

QoS Quality of Service

RAN Radio Access Network

RIS Reconfigurable Intelligent Surface

RL Reinforcement Learning

RNI Radio Network Information

SDN Software Defined Network

SDO Standard Development Organization

UAV Unmanned Aerial Vehicle

UE User Equipment

UL Uplink

URLLC Ultra Reliable Low Latency Communications

UPF User Plane Function

VIM Virtualization Infrastructure Manager

VNF Virtual Network Function

VNFM Virtual Network Function Manager

VRU Vulnerable Road User

V2I Vehicle to Infrastructure

V2V Vehicle to Vehicle

V2X Vehicle to Everything

XR Extended Reality

WPT Wireless Power Transfer

5GC 5G Core network

5GAA 5G Automotive Association

5G ACIA 5G Alliance for Connected Industries and Automation

1. INTRODUCTION

One of the main features of 5G and future cellular systems is the possibility of supporting novel industrial verticals (e.g., by offloading computation tasks or letting run applications into more powerful servers) [1]. However, those use cases could have stringent requirements in terms of optimal Quality of Experience (QoE) and Quality of Service (QoS) and therefore the Cloud Computing paradigm alone cannot support those verticals anymore. For instance, some verticals require a tight latency deadline for a smooth experience (e.g., AR/VR), which is difficult to achieve if processing servers are located in a data center thousands of kilometers away from end users. One way to support these verticals is to leverage edge computing, a decentralized computing model, that positions computational resources closer to the end-users promising enhanced performance, reduced latency, and increased throughput.

This thesis fits into this domain and by leveraging standardized paradigms proposed by 3GPP and ETSI, it tries to find efficient solutions to dynamically support verticals at the edge while at the same time keeping an eye on sustainability (both in economical and energy-efficiency terms) for network operators. In detail, this thesis supports novel verticals by developing intelligent algorithms (e.g., heuristics and DRL-based solutions) for the allocation and migration of computing-demanding tasks in edge servers dependable on intermittent renewable sources, with the ultimate goal of maximizing revenues for network operators while leveraging as much as possible the presence of renewable sources (i.e., decreasing overall costs).

Moving smaller computational resources from the cloud to the edge is not a completely new idea, since for instance several years ago the Fog Computing paradigm was proposed [6], with Cloudlets being a possible application [7]. It is also possible to argue that the more general idea itself stems from Content Distribution Networks (CDN) in 1999 [8]. However, it is with the first rollout of 5G networks, combined with standardization efforts (e.g., ETSI with the so-called MEC paradigm [9]), that edge computing resources are deployed and seamlessly connected to a cellular network ecosystem [10]. However, from the network operator's point of view, realizing the full potential of edge computing requires a meticulous exploration of technical challenges and innovative solutions, while at the same time committing to new compelling general goals.

In the following Section, we describe the novel challenges this thesis will consider and how it will fill the gap with respect to the state-of-the-art. Finally, in Section 1.2 we describe the contributions and impact of each of the publications supporting this thesis.

1.1. Thesis' challenges and contribution

This thesis addresses practical challenges starting from showing bottlenecks and constraints of supporting use cases from an edge infrastructure point of view to defining novel sustainable scenarios and proposing algorithms that take into account the dynamicity of an edge network (e.g., variable presence of green energy, variable workloads among the others).

Let us now briefly analyze below the major challenges we considered in this work:

- *Supporting disruptive verticals*: A novel computing infrastructure deployed at the edge of the cellular network could help grow several industrial verticals, opening the same new possibilities of revenues for network operators. However, each of these verticals has several divergent requirements in terms of QoS and QoE (e.g., automotive has different QoS constraints compared to smart factories or video streaming). Therefore it is important to address possible bottlenecks, in terms of bandwidth, computing capabilities, and reliability.

- *Sustainable edge*: Network infrastructures, should nowadays become more and more sustainable in terms of carbon footprint, a goal which is indeed one of the key pillars for future 6G networks [11], [12]. Expected beyond 5G and 6G use cases such as cloud gaming and AR are computational and energy-hungry. Therefore a sweet spot should be found to support those novel use cases at the edge while intelligently leveraging the available presence of green energy at edge sites. On the same line, sustaining a novel edge infrastructure should also be economically profitable for network operators.

- *Flexible edge provisioning*: Novel networking concepts such as network slicing, Software Defined Networking (SDN), and NFV give a novel flexibility on how to allocate resources at the edge of the network. As we know, edge capabilities are scarce compared to the cloud and therefore resources should be carefully allocated. Timely migration of resources between close edge servers could help the edge paradigm to achieve a better usage of resources (i.e., decreasing monetary costs) and a decreased carbon footprint, leaving room for novel strategies and algorithms.

We now deeply focus on each of these challenges, showing limitations from the recent related literature and commenting on the thesis' contribution for each of the aforementioned challenges. Next, we mention the publications supporting this thesis and how they contribute to the overall work.

1.1.1. Supporting disruptive verticals

One of the goals of future cellular networks is to support the growth of industrial verticals such as IoT, Automotive, Augmented/Virtual Reality, eHealthcare, Media, Smart Factories, and Smart Cities. However, the realization of these applications introduces challenges due to their stringent and divergent requirements. For instance, cloud gaming and AR demand ultra-low latency and significant computing capacity, placing strain on the finite resources available at the edge. Besides, it is already been demonstrated [13] that VR will not be supported in 5G networks and it will be difficult even for the next cellular generation. Furthermore, it is well known that computing edge resources are scarce, and therefore efficiently allocating and managing these resources is critical to ensuring a seamless and responsive user experience. Addressing these challenges brings NFV to the forefront and opens novel opportunities for research.

NFV, thanks to the flexibility given by the software and Virtual Networks Functions (VNFs) enables the dynamic provisioning and management of edge resources, allowing for real-time adjustments based on specific application requirements (a deal-breaker for some verticals [14]). This flexibility ensures that edge resources are optimally utilized, meeting the diverse needs of latency-sensitive verticals and meanwhile minimizing network operators' resource wastage keeping an

eye on sustainability. Furthermore, nowadays Artificial Intelligence (AI) and especially Machine Learning (ML) techniques are becoming more advanced, with the possibility of exploiting them in a cellular networks context (as is also happening in other contexts such as Open-RAN [15]. Techniques such as supervised Machine Learning or DRL could indeed help network operators intelligently allocate and manage their edge resources in an automated manner. According to our survey [1] (and we will discuss it more in Chapter 2), we discovered that some verticals have not been fully evaluated yet even though they are considered important use cases for future-generation cellular networks. In this thesis, we focus our attention on two particular verticals: cloud gaming (especially on its edge version) and Mobile Augmented Reality (MAR). Both verticals represent a growing market [16], [17] and their tight requirements (e.g., computing and latency) require particular attention from network operators to both guarantee a pleasurable QoE for end users and a good utilization of edge network resources. Network operators could support those verticals by for instance allowing the computing of online gaming sessions at edge servers, or by allowing the offload and computation of MAR tasks. We now present an overview of related works with an emphasis on cloud gaming, edge gaming, and MAR tasks offloading, and later on we comment on the thesis contributions in these topics.

Cloud gaming: Cloud gaming has been well studied and is becoming a pervasive use case, with some projections highlighting that 20% of gaming sessions will be soon on cloud [18]. Several recent papers addressed this paradigm, focusing especially on: *i)* server allocation, *ii)* costs and *iii)* QoS guarantees. In [19], the authors study the server provisioning problem for cloud gaming with the double goal of reducing server running and software storage costs. Similarly, the authors in [20] propose several heuristic algorithms to solve the problem with both server allocation costs (for server renting fees and data transfer) and the bandwidth costs, taking into account real-world latency constraints, while Wu *et al.* [21] design an online control algorithm to reduce both latency (focusing especially on queuing delay) and server provisioning costs. Some works study resource utilization, using real testbeds: Li *et al.* [22] focus on minimizing resource usage when interference between co-located games at the same server happens (i.e., decreasing QoS). According to [14], games could fall into two categories: CPU-critical and memory-input-output-critical. Therefore, they propose several task scheduling strategies to optimize resource allocation. In [23], the authors propose a framework called T-Gaming that uses off-the-shelf consumer GPUs, prioritized video encoding, and adaptive real-time streaming based on deep reinforcement learning to reduce hardware and network costs. Other works analyze the resource placement problem. For example, Hong *et al.* [24] study the Virtual Machine (VM) placement problem for maximizing both the profit for service providers and the overall gaming QoE for end-users while in [25], the authors propose a distributed algorithm to optimize VM placement in mobile cloud gaming through resource competition.

Edge gaming: Compared to cloud gaming, edge gaming is a newer concept tied up with edge computing. Indeed, edge computing could help the cloud gaming paradigm in both storage [26] and computation (by offloading tasks [27] or rendering whole games), minimizing the overall response time. In [28], the authors leverage edge servers to offload computation-intensive tasks for gaming, showing that this strategy could reduce network delay and bandwidth consumption. Yates *et al.* [29] develop a Markov model to optimize frame rate and lag synchronization of server and player in low-latency edge cloud gaming systems, employing an age of information metric to characterize the system performance.

MAR: The problem of allocating MAR tasks in an edge scenario has been extensively studied in the literature from different perspectives. For example, [30] studies this problem on a multipath edge network, Ren *et al.* [31] propose a three hierarchical MEC-based computation framework for supporting AR, and in [32] the authors design an edge network orchestrator trading off between low latency and accurate object analytics. [33] and [34] provide a detailed implementation on how to integrate ETSI MEC and 5G networks with MAR. Many other papers in this domain considers also the energy efficiency problem on the end-device side. We will comment on some of them in the following subsection.

Thesis Contribution: To summarize, while cloud gaming and MAR topics have already been covered by the research community in different flavors, edge gaming is still in its early steps, therefore leaving room for more research studies. This thesis focuses on supporting these two use cases (edge gaming and MAR) in a cellular network while also considering the sustainable aspects (both in economical and carbon footprint terms) at the edge infrastructure side, which has not been evaluated yet in these two domains.

1.1.2. Sustainable edge

The deployment of an edge computing infrastructure requires substantial investments in hardware, connectivity, and ongoing maintenance [35]. Striking the right balance between cost-effectiveness and scalable infrastructure is pivotal to ensuring the economic viability of edge computing. A strategic approach to managing both CAPEX and OPEX is essential for the successful deployment and sustainability of edge computing solutions. Nowadays, many interesting solutions and paradigms have been proposed and deployed. For instance, with NFV [36] the softwarization of network functions is possible, with the benefit of using general-purpose hardware instead of specialized one. Then, standardization efforts are becoming more and more predominant in research and industry (ETSI MEC [9], Open-RAN [37] among the others), thus paving the way for a common hardware and interface infrastructure between different players. These solutions could help network operators in decreasing monetary costs when deploying novel infrastructures. Moreover, a novel edge infrastructure could open the door for additional sources of revenue for network operators (e.g., by leasing their edge computing capacities to verticals or users) but, at the same time, over-utilizing edge resources could lead to using more energy (i.e., an increased electrical bill (OPEX costs)) and a decrease of QoE for end-users.

We also live in an era dominated by environmental concerns and the carbon footprint associated with infrastructure operations, demands urgent attention. Indeed, the United Nations issued several Development Goals [38] and among them, there is the goal of building and promoting sustainable industrialization. On the same level, the European Commission issued the EU Green Deal, and the research community agreed that future 6G networks should employ energy-efficient techniques. Edge computing deployments must align with sustainable practices, and network operators should emphasize energy-efficient models/techniques and incorporate renewable energy sources (or use energy generated from these sources, the so-called *green* energy) into their infrastructure. Indeed, it has been demonstrated that leveraging on energy efficient techniques could help network operators save more money, thanks to decreasing general OPEX costs [11], [12], [39]

More in detail, in our survey (and also reported in Section 2.2, we show that CAPEX and OPEX costs will rise for sustaining a pervasive edge infrastructure. While the numbers of Figure 2.11 represent a future cellular generation scenario, since nowadays only a few MEC nodes (where edge computing resources are placed) have been deployed in a metropolitan context, however, the monetary costs are non-negligible. Since there is a growing awareness of making infrastructures more sustainable to decrease the overall carbon footprint, a possible solution to making the edge more sustainable would be to leverage on intermittent renewable sources (e.g., solar/wind) to generate green energy close to edge servers. In this way, edge servers could benefit from the presence of *free* or zero cost green energy to power their edge system and in case only take costly *brown* energy (e.g., generated in carbon-fuel power plants) from the network grid when required. This could be an option for network operators to decrease the overall carbon footprint while also reducing OPEX costs.

Looking at the literature, there are not many papers that consider the presence of green energy on the server side in cloud/edge gaming and MAR use cases. At a high level, only the authors in [40] discuss green energy solutions for cloud gaming while in [41], Chuah *et al.* propose a control algorithm to decrease GPU power consumption while guaranteeing the Service-Level Agreement (SLA). Instead, of focusing on MAR, there also exist works that focus on the interplay of edge offloading for MAR tasks and energy efficiency with the use of ML techniques, which are consequently closer to our thesis' scope. For instance, Chen *et al.* [42] minimize the energy consumption of each user when offloading MAR tasks to MEC. In a similar scenario, the authors in [43], leveraging Deep Learning techniques, propose an energy-efficient task offloading algorithm to minimize the battery consumption of devices. Always leveraging DRL, Chen *et al.* [44] propose an AR tasks offloading scheme that maximizes the computation rate and energy efficiency in Beyond 5G systems, and Wang *et al.* [45] design an energy-aware system that enables MAR clients to dynamically change their parameters to minimize their per-frame energy consumption. Furthermore, Cheng *et al.* [46] study AR task delay and power consumption minimization, while in [47] the authors leverage DRL to reduce the overall (transmission and server computation) energy cost while meeting the latency requirements. Ahn *et al.* [48] propose a theoretical framework to improve both the resolution of offloaded frames and the energy efficiency of multiple MAR devices connected to a single MEC server. In [49], the authors configure the AR tasks offloading problem as a partially observable Markov decision process to minimize the energy consumption of mobile devices while guaranteeing the deadlines of real-time tasks.

However, due to the unpredictability of renewable sources (e.g., sun/wind [50]), network operators cannot completely rely on the presence of green energy generation, and at the same time, they should be able to serve as many users as possible to increase their revenue streams (e.g., by leasing their edge servers to AR applications for offloading computing tasks). Therefore, a network operator should be able to create strategies that help them in both benefiting the presence of green energy as much as possible but also maximizing serving end-users. More in detail, operators should be able to strategically place workloads based on the availability of renewable energy, to reduce OPEX costs while contributing to environmental sustainability. Due to the intermittence of this energy generation, network operators should also consider the importance of dynamic resource (or task) migration between close edge servers, which could help in optimizing the allocation of edge resources based on evolving goals and demands. Migrations could facilitate load balancing and scalability. Indeed, during peak periods, resources can be redistributed

to handle increased workloads, preventing bottlenecks and maintaining consistent performance. Instead, during low-demand periods, resources can be efficiently consolidated to minimize energy consumption. Especially for some verticals (e.g., cloud gaming and AR), where latency and computational capacity are critical, dynamically migrating tasks between edge servers becomes essential for maintaining optimal QoS and QoE for end-users. To conclude, migrating resources based on the presence of green energy sources at different edge locations offers the dual benefit of *(i)* decreasing operational costs by tapping into renewable energy but also *(ii)* aligning with environmentally conscious practices.

Thesis Contribution: As discussed above, in literature, not many papers consider the presence of green energy given by intermittent renewable sources at edge servers side in both edge gaming and MAR use cases. This thesis fills this gap by proposing and analyzing scenarios where edge servers partially or completely depend on intermittent renewable energies while they have to support computation and energy-hungry use cases. On the same line, edge operators should consider revenues and monetary costs, to make the edge infrastructure profitable. We remark that this sustainable scenario has not been fully evaluated yet and represents a clear novelty to the literature.

1.1.3. Flexible edge provisioning

The problem mentioned above (finding a sustainable sweet spot between leveraging as much as possible of green energy while keeping up a high revenue stream) is conflicting. Indeed, a network operator wants to maximize the acceptance of users' tasks in their edge system (bringing new revenues to them) but at the same time it wants to decrease the OPEX cost of using brown energy, which means that if there is a little presence of green energy, the acceptance of users tasks should decrease. In this dynamic scenario, a network operator should be able to dynamically allocate and migrate users' tasks across several edge nodes, to follow the presence of green energy, reducing the impact of using costly brown energy and therefore accepting as much as users' tasks (which bring more revenues in the system).

Allocation and migration of resources is a well-known problem, which has been studied especially for centralized cloud systems [51]. Notwithstanding, edge nodes could be constrained for instance in bandwidth, energy, storage, or computing capabilities, and it is difficult to apply the same strategies designed for cloud systems into an edge scenario due to these multiple constraints on the edge resources. Among the works that consider this challenge in edge networks, [52] proposes a joint service placement and request scheduling scheme, while [53] proposes a randomized rounding technique for the joint optimization of service placement and request routing in a MEC network. Both papers consider several constraints on edge nodes. Other papers provide solutions from a different perspective, and they make use of a machine learning approach; for example, Wang *et al.* [54] propose a joint task offloading and migration schemes in a mobility-aware MEC network. Their scheme is based on Reinforcement Learning (RL) and their goal is to obtain the maximum system utility minimizing the migration costs. In [55], the authors use a deep learning framework for proactive migration (based on service replication) of MEC resources in a 5G vehicle scenario, to minimize the total energy expenditure, without considering hardware limitations such as on memory and CPU cycles. In [56], the authors leverage on deep reinforce-

ment learning to minimize the average completion time of tasks under migration energy budget, while in [57] the authors investigate the task migration issue for multiple Unmanned Aerial Vehicles (UAV)s in the MEC-based UAV delivery system. Specifically, they study an energy-aware decision-making strategy for dynamic task migration to optimize the UAV energy consumption. Wang *et al.* [58] propose a Markov decision process framework for dynamic service migration to follow users' movement, without considering edge nodes' capabilities. In [59], the authors propose a resource-aware VM migration technique, without taking into account energy consumption, while the authors in [60], on the opposite, focus on a VM migration mechanism that is aware and adapts to the fluctuating available green energy, minimizing therefore the energy consumption from non-renewable sources, but without considering other constraints. Braun *et al.* [61] propose a new migration protocol to migrate a MEC gaming application through different edge servers while in [62] the authors have developed Talaria, an in-engine content synchronization solution. The latter allows for unnoticeable game instance migration between edge servers, which is mandatory to maintain a satisfactory QoE for end-users in fast-paced games. Finally, many works consider the broader topic of edge allocation for generic tasks using DRL [63]–[65].

Thesis Contribution: None of the papers consider the allocation and migrations of resources in the scenario of edge gaming or MAR offloading tasks with edge servers that could depend on intermittent renewable energies. In this thesis, we propose two different allocation and migration strategies for both verticals. In particular, for edge gaming, we propose an intelligent heuristic that greedily allocates and migrates gaming tasks according to the variable presence of available green energy. Instead, for the more challenging MAR use case, we leverage machine learning techniques, and in particular DRL, to build a DRL-based solution. In our thesis, we show that both approaches obtain performance close to optimal ones and have a clear improvement compared to state-of-the-art approaches.

1.2. Research output and publications

Driven by all the above observations, in this thesis we present different ideas and solutions to solve the aforementioned problems. The main contributions of this thesis have been published in 4 publications. More specifically, 1 has been published in IEEE Communications Surveys & Tutorials (indexed in Journal Citation Reports (JCR)) and 1 in Elsevier Computer Communications (indexed in JCR). Moreover, 1 publication has been published at ACM MSWiM 2023 and another one at WoWMoM '22, tier A and B according to CORE2014 datasets. In details,

Contribution 1. *Extended literature review on MEC, with a special focus on how supporting verticals at the edge.*

First of all, we provide a literature review of the MEC paradigm, in particular focusing on three aspects: *(i)* we first devise the efforts for standardizing MEC and in particular we focus on the work done by ETSI. Next, we *(ii)* focus on the flexible provisioning of edge resources, focusing on NFV techniques and migration procedures. Afterwards, we *(iii)* highlight how verticals (IoT, Automotive, (AR/VR, eHealthcare, Entertainment, Smart Factories, and Smart Cities) could leverage MEC and show what has been done and open research challenges. Finally, we studied a smart metropolitan example which helped us in understanding bottlenecks and difficulties in supporting several verticals in a smart metropolitan context. Here we give a glance at the expected

costs for deploying and maintaining a pervasive edge infrastructure and how several topics have been not extensively tackled yet in the literature (especially some particular verticals) and how network operators could leverage green energy at edge servers side.

- **F. Spinelli** and V. Mancuso, "Toward Enabled Industrial Verticals in 5G: A Survey on MEC-Based Approaches to Provisioning and Flexibility," in IEEE Communications Surveys & Tutorials, vol. 23, no. 1, pp. 596-630, Firstquarter 2021, doi: 10.1109/COMST.2020.3037674.

Contribution 2. *Study of the green edge gaming scenario and proposal of a smart heuristic called* `GREENING`*.*

Based on open research questions we got from our survey, we developed and studied an innovative concept called *green edge gaming*. This concept describes a scenario where online gaming sessions are allocated in an edge network, with edge servers partially or completely depending on green energy. As mentioned before, due to the high variability of renewable sources and the scarce computing capacities of edge servers, it is important to timely allocate and migrate resources while maintaining a high acceptance rate of gaming sessions in the network, which could lead to a new stream of revenue for network operators. We formulate an optimization problem and we propose a smart heuristic, `GREENING`, which allocates and migrates gaming resources depending on the presence of green energy.

- **F. Spinelli** and V. Mancuso, "A Migration Path Toward Green Edge Gaming," 2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), Belfast, United Kingdom, 2022, pp. 347-356, doi: 10.1109/WoWMoM54355.2022.00033.
- **F. Spinelli**, Antonio Bazco-Nogueras, and Vincenzo Mancuso. "Edge Gaming: A Greening Perspective." Computer Communications 192 (2022): 89-105.

Contribution 3. *Present a DRL-based solution to find a compromise (with a proportional fairness structure) between incoming revenues and the decreasing of carbon footprint.*

We present another different edge scenario, in which AR computing tasks have to be offloaded into an edge network. In particular, we show how the problem of accepting tasks and using green energy as much as possible is conflicting and therefore we formulate an optimization problem with a proportional fairness structure that helps us in finding a compromise between these two goals. For this problem, we leverage DRL, since ML techniques are becoming more and more important for the automation of network management, due to both the rising complexity of the network and their power to address highly complex problems that involve large amounts of data. Therefore, we propose `GreenRL`, a DRL-based solution that intelligently offloads and migrates AR tasks according to the presence of green energy at the edge servers side.

- **Francesco Spinelli**, Antonio Bazco Nogueras, and Vincenzo Mancuso. 2023. Offloading Augmented Reality Tasks with Smart Energy Source-Aware Algorithms at the Edge. In Proceedings of the Int'l ACM Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM '23). Association for Computing Machinery, New York, NY, USA, 73–82. https://doi.org/10.1145/3616388.3617523.

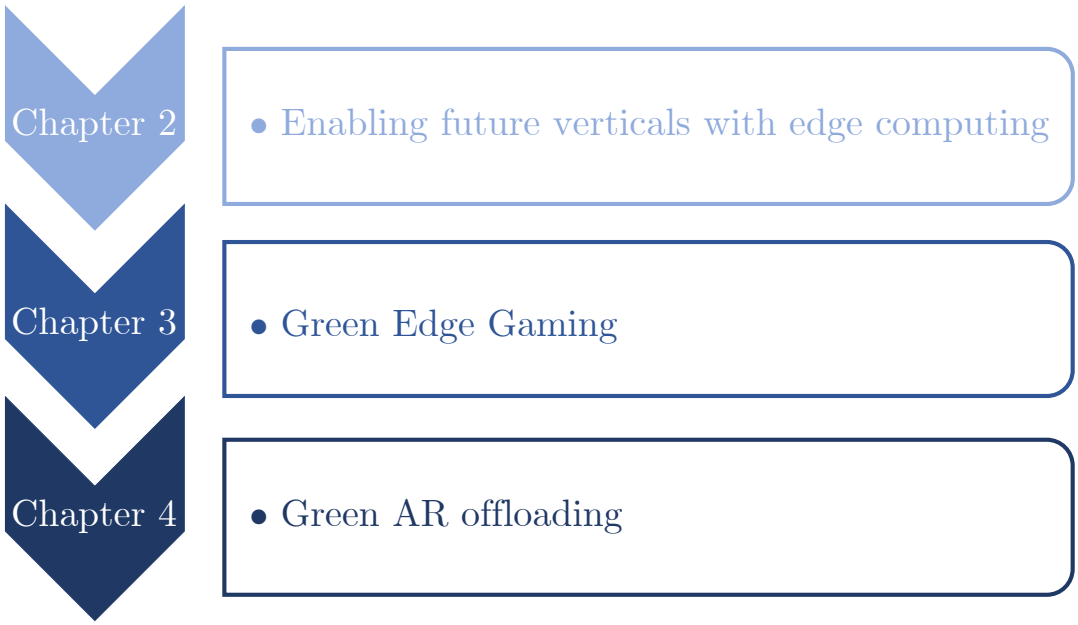


Figure 1.1: High-level illustration of the thesis structure.

1.3. Outline of the thesis

The rest of the thesis is organized in different chapters detailing the contributions aforementioned in the previous subsection, as shown in Fig. 1.1.

In Chapter 2 we perform an extended literature review of MEC. In particular, we focus on three different aspects of MEC: the standardization process, in Section 2.1.1, the flexibility given by NFV (Section 2.1.2) and how industrial verticals can leverage on the MEC presence (Section 2.1.3). Next, in Section 2.2 we also present a study where we focus on bottlenecks and issues in supporting several verticals simultaneously in a smart metropolitan context. Finally, in Section 2.3 we conclude the Chapter.

In Chapter 3 we analyze a novel concept we developed (*green edge gaming*), or how to support gaming at the edge with edge servers that depend on green energy. In Section 3.1 we motivate our concept; in Section 3.2 we present the system model. In Section 3.3, there is the formulation of an instantaneous optimization problem and we prove that it is NP-hard in the strong sense. Next, in Section 3.4 we extend the problem in time and propose an efficient heuristic to solve while in Section 3.5 we analyze the performance of our heuristic in several scenarios. In Section 3.6 we summarize the Chapter

Afterwards, in Chapter 4 we leverage machine learning applications to support the smart offloading of AR tasks at the edge maximising the presence of green energy. In particular, in Section 4.1 we overview our scenario of green AR offloading and in Section 4.2 we show the system model of the study. Next, in Section 4.3 we formulate our optimization problem and we show that it is NP-hard. In Section 4.4 we present our solution based on Deep reinforcement Learning solution, in the next Section 4.5 we show the results and in Section 4.6 we summarize the Chapter.

Finally, in Chapter 5 we present the concluding remarks of the thesis and we identify possible future research lines that arise from our work.

2. ENABLING FUTURE VERTICALS WITH EDGE COMPUTING

The increasing number of heterogeneous devices connected to the Internet and tight 5G requirements have generated new challenges for designing network infrastructures. Industrial verticals such as Automotive, Smart City, and eHealthcare (among others) need secure, low latency, and reliable communications. To meet these stringent requirements, computing resources must be moved closer to the user, from the core to the edge of the network. However, the deployment and provisioning of edge resources come with added challenges for network operators. In this Chapter, we give an extended and updated literature review of some of the most challenging aspects of MEC and in particular we answer three research questions about the MEC as a means to deploy virtualizable edge computing services in mobile networks: *(i)* how?, *(ii)* where?, and *(iii)* for what? For the first question, we overview the ETSI efforts for MEC standardization. Indeed, vertical industries' use cases in 5G and 6G networks impose major architectural changes to mobile networks to simultaneously support a diverse variety of stringent requirements. Therefore, it is important to develop a standardized, open environment for efficient and seamless integration of applications from vendors, service providers, and third parties across multi-vendor computing platforms at the edge of mobile networks. In Section 2.1.1 we briefly overview the ETSI MEC architecture and its integration with 5G networks, showing that thanks to NFV, the MEC architecture could be deployed in 5G networks in flexible ways, thus leaving room for novel research scenarios. Finally, we comment on recent efforts from ETSI and researchers to contribute to the MEC standardization process. For what concerns the second question, in Section 2.1.2 we build upon what we discussed in the previous Section and we focus on MEC provisioning features within 3GPP network architectures, giving a glance of possible techniques and issues in supporting verticals with MEC. In particular, we focus on computation offloading, how and where to deploy MEC resources, and the agile migration of MEC-VNF-based resources. Finally, in Section 2.1.3 we explore how the MEC can be used to deploy online services and specifically how it will enable several industrial verticals in 5G and future cellular networks. In Section 2.2, we give a high-level example of MEC deployment in a smart metropolitan area supporting multiple industrial verticals at the same time, exposing bottlenecks and constraints. Eventually, we comment on how the findings present in this Chapter shaped our following works and therefore this thesis.

2.1. Enabling Multi-access Edge Computing

According to ETSI, MEC is *IT service environment and cloud-computing capabilities at the edge of the mobile network, within the Radio Access Network (RAN) and close to mobile subscribers* [66]. Examples of MEC applications include caching of DNS entries, caching of contents to deliver to customers, and tracking of devices. Furthermore, there are many proposals about using the MEC for implementing advanced network functions, e.g., for enhanced secure VPNs, as well as for computational offloading and collaborative computing purposes, indoor localization, distributed data analytics, assisted driving and control of vehicle platoons, smart infotainment with adaptive video transcoding and support for augmented and virtual reality, control of smart grids, and support for IoT and smart environments in general (smart cities, smart factories, smart health

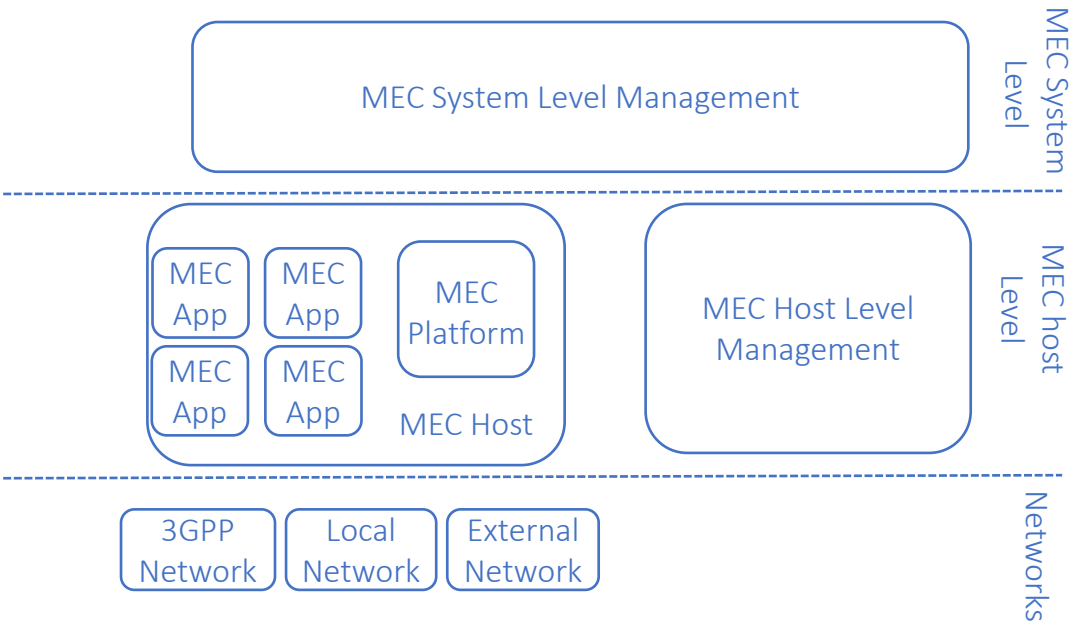


Figure 2.1: ETSI MEC Framework.

care systems, etc.).

MEC has been standardized by ETSI, which created a MEC Industry Specification Group (MEC ISG) and published the first white paper in September 2014 [67]. ETSI has released and updated more white papers and technical specifications in the following years. A new technology standardization process is very important for many reasons: *(i)* it allows interoperability between products, *(ii)* to brainstorm and clarify the challenging technical aspects and *(iii)* merge technical solutions with research advancements.

2.1.1. How: Standardization

Fig. 2.1 illustrates the general entities involved in the MEC architecture, according to ETSI [9]. Three different levels are present: the upper one is the MEC *System Level*, which has a global visibility on the MEC architecture and therefore coordinates every block in the levels below. In the middle, the MEC *host level* includes MEC host and MEC host level management. The MEC host is an entity that includes the platform and the virtualization infrastructure used to run the MEC, and which provides network resources, storage, MEC services, and computing power for MEC applications. MEC services are provided and consumed by MEC applications or the MEC platform. Some examples are the Radio Network Information (RNI), which gives information on the radio network state, the location service, which gives location-related information and the bandwidth manager service, which helps prioritize and handle traffic. Containers or virtual machines run as well in the MEC host and can leverage MEC services. At the bottom of the stack, Fig. 2.1 shows various transmission entities such as 3GPP cellular networks and local/external networks. This shows that the MEC will be able to support many access technologies, even at

the same time, giving the possibility to exploit fixed mobile convergence[1], which is a 5G feature meant to allow devices to connect through both wired/wireless transmissions at the same time.

Figure 2.2 shows all the most important elements contained inside the MEC reference architecture, and the reference points connecting the whole system. Reference points are divided in 3 different categories:

- *Mp* are the reference points located inside a MEC platform, allowing the connectivity between MEC platforms, MEC applications and the data plane.
- *Mm* reference points are instead for management purposes.
- *Mx* reference points connect MEC elements towards external entities.

Describing the MEC system from the top (hence from the *system level*), requests to the MEC infrastructure are sent in two different ways: with a User Equipment (UE)/Device Application, or through a Customer Facing Service (CFS) portal. The latter allows operators' third parties to select a set of MEC applications given their needs and it is directly connected to the Operations Support System (OSS) through the *Mx1* reference point. Instead, from the UE, the requests are first sent through a *Mx2* reference point to the User application life cycle management proxy. This entity checks if the requested application is already instantiated and, otherwise, it forwards the request to the OSS. Moreover, it also informs the UE about the state of the application and it supports applications relocation inside or outside the MEC system. It is connected to the OSS through the *Mm8* reference point, and to the Multi-Access Edge Orchestrator (MEO) via the *Mm9* reference point.

The OSS receives the requests from both the CFS and the proxy and determines request granting, sending the requests to the MEO in positive cases. The OSS leverages the *Mm1* reference point, which triggers the instantiation and the termination of MEC applications, and on the *Mm2* to connect with the MEC platform manager. Furthermore, the OSS gives the possibility, upon device request, to relocate MEC applications to external clouds. The last element of the *system Level* is the MEO. It maintains an overall view of the MEC system, knowing the available resources, services, and deployed MEC hosts, and it also monitors the topology. It selects the best host where to deploy an application, taking into account available resources, services availability, and constraints such as latency. Moreover, it is responsible for operator policies and it interfaces with the Virtualization Infrastructure Manager (VIM) for preparing the physical infrastructure. It is connected with the MEC platform manager via the *Mm3* reference point, for application life cycle management and for keeping track of the available MEC services, and with the VIM.

Three different entities are present in the MEC *host level*: the MEC host, the VIM, and MEC platform manager. The latter is responsible for managing the life cycle of both applications and MEC platforms, and for receiving information on faults and performance measurements from the VIM, hence informing the MEO if any relevant event happens. The MEC platform manager is connected to the VIM via the *Mm6* reference point and to the MEC platform via *Mm5* reference point, allowing the platform configuration and applications life cycle procedures. The VIM allows the management of the virtualization infrastructure located inside the MEC host, managing the

[1]https://www.telefonica.com/en/web/press-office/-/telefonica-presents-the-first-prototype-of-an-open-and-convergent-access-network-that-integrates-fixed-and-mobile-and-enables-edge-computing

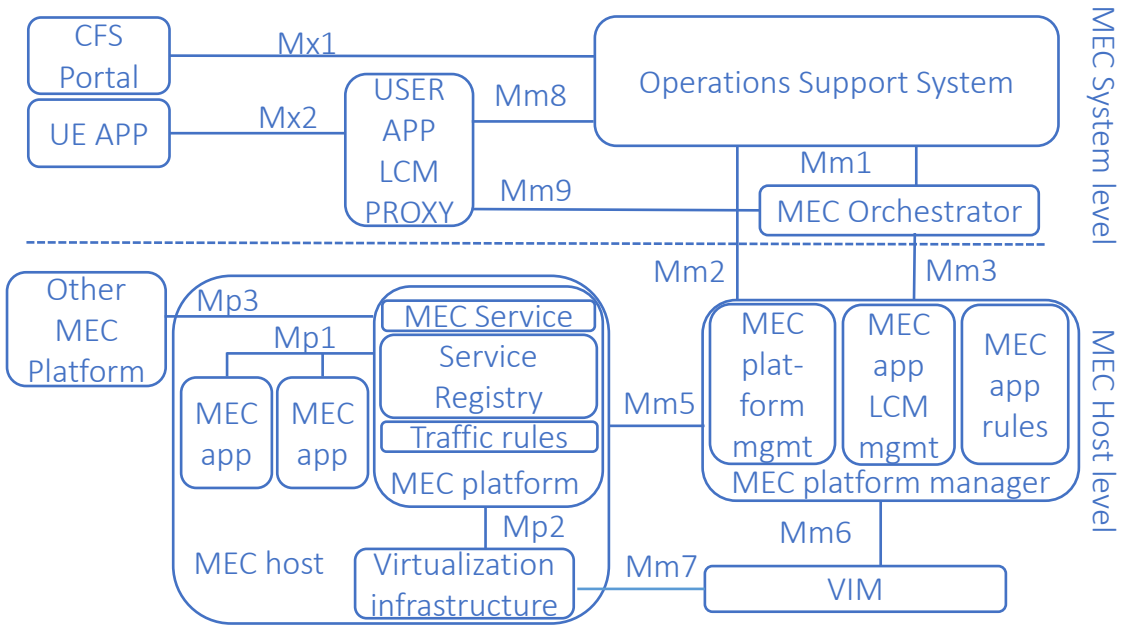


Figure 2.2: MEC Architecture.

allocation and release of virtualized resources, preparing the infrastructure to run a software image and it supports the rapid provisioning of applications, as described in [68]. It is connected with the virtualization infrastructure through the *Mm7* reference point. The MEC host is further divided into three different sub-entities:

- Virtualization infrastructure, which provides the computing and network resources and the data plane.
- MEC platform, which offers its services to the applications and talks with other MEC platforms under the same MEO; moreover, the MEC platform receives traffic rules and DNS configurations from the MEC platform manager and instructs the data plane following those rules.
- MEC applications are deployed as virtual machines or containers on top of the virtualization infrastructure. They interact with the MEC platform, providing the required services or leveraging on already instantiated MEC services and management information. Services hence are placed either inside the MEC applications or in the MEC platform, meaning they are directly deployed and controlled during the MEC platform instantiation.

As an example, Fig. 2.3 shows a possible ETSI MEC framework deployment supporting platooning of assisted-driving vehicles. Thanks to its distributed architecture, the MEC host is deployed at the network edge, near the base station (the *gNB*, using the 5G jargon), while MEO and OSS, which need a more centralized view, can be deployed more inside the network. The MEC will provide support for platooning by storing, updating, processing, and sharing information about road traffic, handling requests to join or leave the platoon, or helping vehicles by offloading part of their computation tasks.

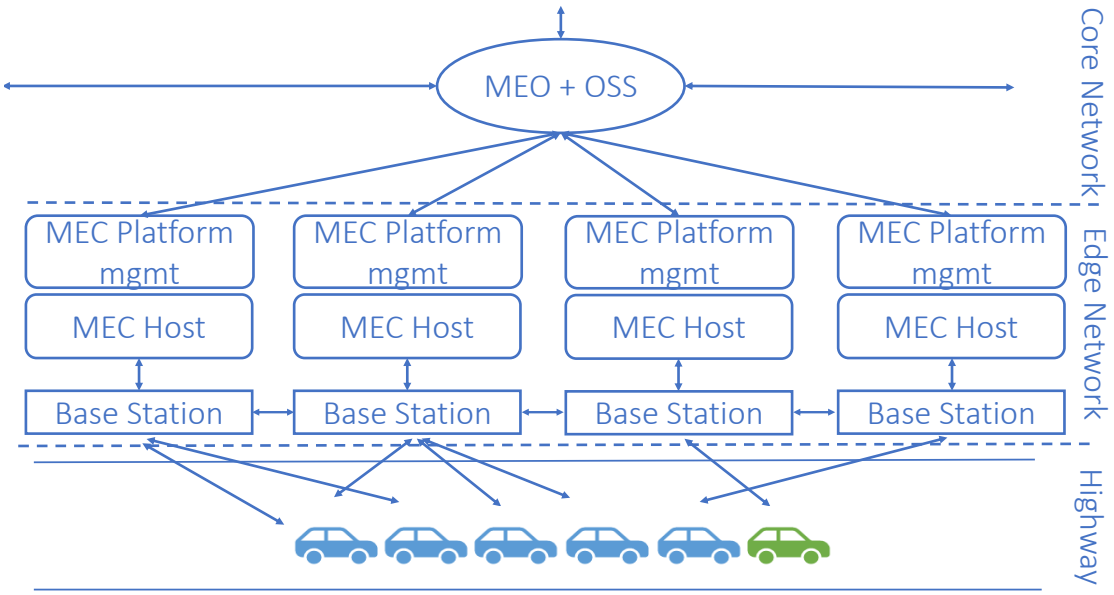


Figure 2.3: MEC deployment in Platooning use case.

MEC and 5G

Together with SDN and NFV, MEC has been a key pillar of 5G since early discussions [66]. 5G networks require tight constraints on bandwidth and latency, achievable only by moving computing resources from the network core to the edge [69]. At the same time, operators are transforming themselves into vendors of versatile service platforms, so that the MEC concept becomes desirable for them [66].

The concept of MEC had been already partially standardized in a 4G context when 5G requirements and the actual design were still in a primordial phase. However, the deployment of MEC in 5G is different from the one for 4G. MEC was an add-on for 4G, which was already deployed when ETSI first introduced the MEC. Instead, 5G has been holistically designed with the MEC [10]. In particular, ETSI standardization efforts are built on top of the 3GPP specifications for 5G systems (such as 3GPP TS 23.501 [70]), allowing therefore the mapping of MEC blocks onto Application Functions (AFs) of 5G.[2] This allows the use of services and information of 5G 3GPP network functions in the MEC. Furthermore, new functionalities have been defined to provide flexible support for several MEC deployments, taking into account MEC support for user mobility.

Fig. 2.4 shows the integration of MEC in 5G. Since this chapter and thesis focus mainly on MEC, the figure only shows 5G network functions needed for MEC deployment. The User Plane Function (UPF) is the most important one. UPF is a distributed and configurable data plane (seen from the MEC perspective), routing user plane traffic to the appropriate Data Network (DN). Its deployment is coupled with one of the MEC hosts, which is either located in the same DN, to achieve low latency and high throughput at the edge, or reachable through the N6 reference point,

[2]Application Functions are logical elements of the 5G architecture defined by 3GPP. They provide session-related information, used to enable the interaction between control-plane Network Functions.

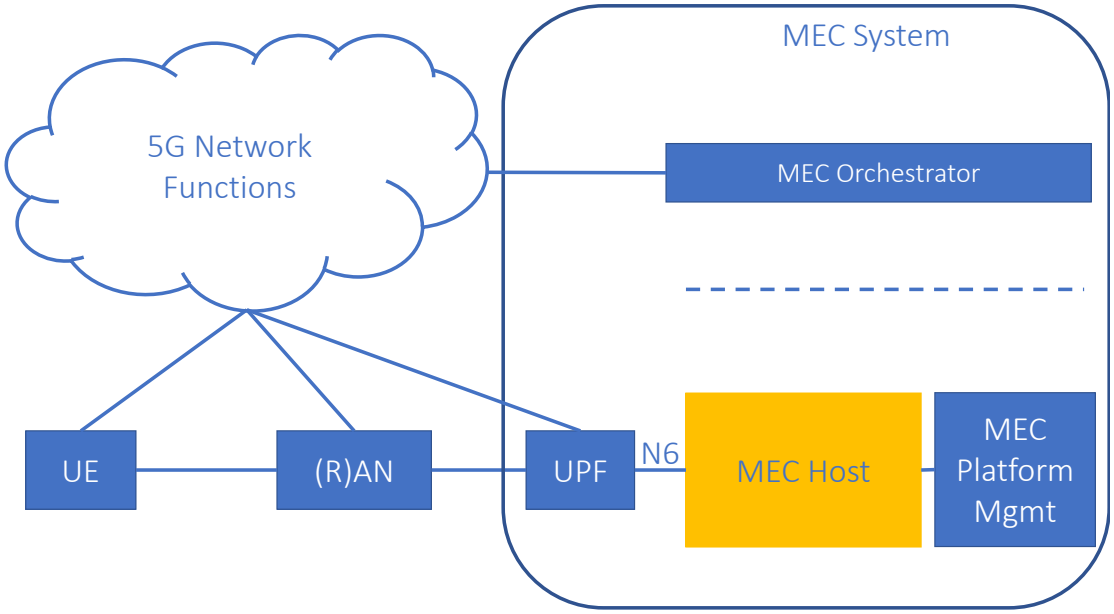


Figure 2.4: MEC with 5G.

which could be external to the 5G system, thanks to the deployment flexibility given by the UPF. Focusing on the MEC control side, the MEC Orchestrator can interact with the 3GPP Network Exposure Function (NEF) or with the target 5G network function[3]. At the MEC *host level*, the MEC platform can interact with the 5G network functions. MEC hosts will be deployed either at the edge or inside the mobile network, even at the core of the network. It is the responsibility of UPF to steer the traffic towards the targeted MEC applications. Moreover, in [70], 3GPP presents the most important enablers for edge computing, which are fundamentals for a correct MEC deployment in 5G networks [10]. These enablers are:

- *Local Routing and Traffic Steering:* The 5G core network architecture allows routing and steering traffic inside the local data network. AFs can also define specific traffic rules.
- *User plane Reselection and Selection:* AFs can define UPF traffic routing and (re)selection. This depends on the UPF deployment scenario and on the configuration of MEC services.
- *(Support of) Local Area Data Network (LADN):* This is enabled thanks to the UPF location flexibility, allowing to deployment of MEC hosts between UPFs and a data network.
- *Session and Service Continuity (SSC):* It allows MEC to fully support user and application mobility.
- *Network Capability Exposure:* Through the NCE, the MEC has indirect access to 5G network functions.
- *QoS and Charging:* This makes it possible to route traffic to a LADN according to the QoS required.

[3]Other 5G network functions are Network Resource Function (NRF) and Network Slice Selection Function (NSSF). For more details, please refer to [10].

Moreover, ETSI has recently published recommendations for the MEC support network slicing [71]. According to recommendations, entities such as MEO, MEC platforms, and MEC platform managers should be aware of slices. Therefore, ETSI proposes to expand the reference points between these entities to include information on network slices. This was revealed to be a very powerful tool. Indeed, based on ETSI recommendations, and on the results presented in [71], Ksentini *et al.* [72] were able to design an ETSI MEC orchestration/management architecture for network slicing, compliant with both ETSI and 3GPP. However, ETSI recommendations for network slicing still have several shortcomings. In [73], the authors addressed those limitations and proposed two solutions for multi-slice MEC support: a Slice Control Function (SCF) to deploy slice-aware MEC App allocation and an inter-slice communication channel to allow the exchanging of data in the same MEC facility.

Fig. 2.5 shows four deployment possibilities of MEC in 5G networks [10]: *(i)* MEC and UPF collocated together with the gNB, *(ii)* MEC deployed with a transmission node, possibly with a local UPF, *(iii)* MEC and local UPF located together with a network aggregation point, and *(iv)* MEC collocated with Core Network functions, inside a data center. The options presented above show how MEC can be flexibly deployed in different locations from near the gNB to a remote data network, which means that, notwithstanding its name, the MEC does not necessarily run at the edge of the mobile network![4] The UPF is deployed and used to steer the traffic towards the targeted MEC applications and the network.

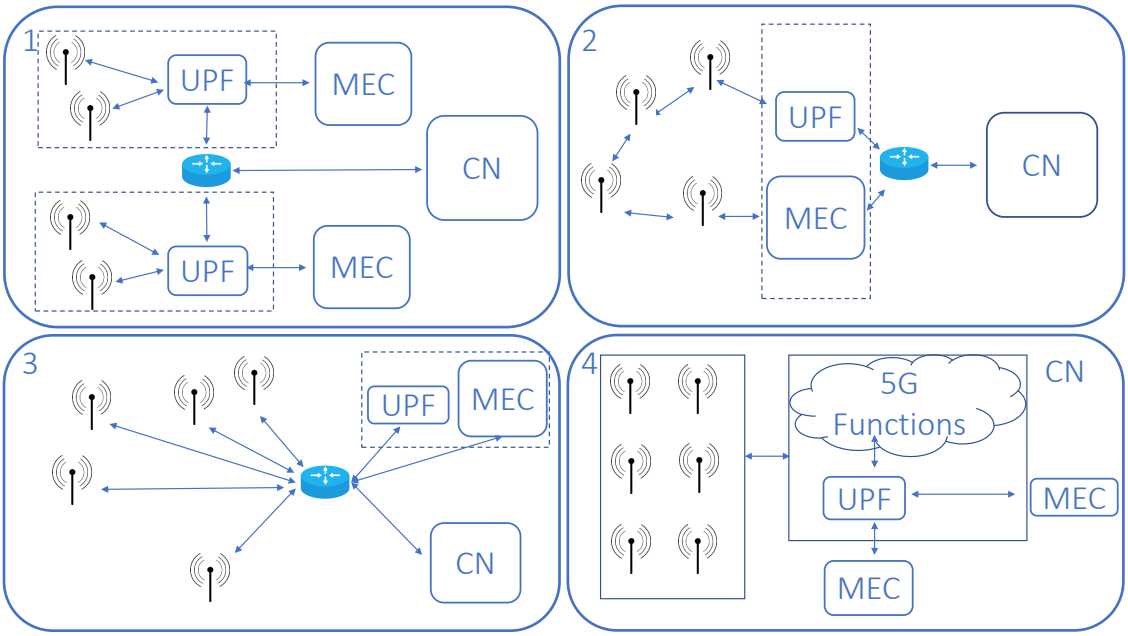


Figure 2.5: MEC deployment scenario in the 5G context.

On the 3GPP side, several technical reports explain how to deploy MEC in 5G networks. For instance, 3GPP SA2 TR 23.748 [74] provides suggestions for several edge computing architecture enhancements in the 5G core network (5GC). The key system enhancements consist in:

- methods to discover the application server IP address at the network edge;

[4]Running MEC hosts far from the edge will be useful in scenarios in which compute power requirements are tighter than latency ones.

- 5GC enhancements to support seamless migration of application servers;
- methods to provide local application servers with network and/or traffic information, in a small amount of time;
- support for traffic steering in an edge N6-LAN.

That document also provides deployment guidelines for use cases such as URLLC, CDN, V2X, AR/VR.

In SA6 TS 23.558 [75], 3GPP specifies the application layer architecture (based on previous 3GPP technical reports), procedures, and information flows needed for a correct deployment of edge applications over 3GPP networks. Further, they provide a first high-level example of how their application layer architecture would merge with ETSI MEC. Finally, in TR 23.758 [76], 3GPP specifically studies architecture requirements for authentication of clients and discovery of edge services, stating that the mapping of those entities and ETSI MEC is considered future work. Therefore, given the flexibility by NFV, novel techniques and approaches should be studied to fully leverage on the presence of MEC in a 5G and beyond 5G networks. In the following subsections (2.1.2 and 2.1.3) we comment on some of these approaches and how MEC can support novel use cases.

Other standardization contributions:

ETSI has also standardized MEC features together with other concepts such as the NFV MANO framework. In this case, the goal of ETSI was to build a MEC system on top of the NFV MANO framework, connecting the MEC entities with the NFV MANO entities [77]. Other efforts consist of integrating MEC and Cloud RAN [78], proposing a MEC deployment in an enterprise environment [79], and in 2018 ETSI published a white paper about MEC supporting V2X uses cases [80]. More recently, ETSI has also considered several novel networking scenarios, taking into account different use cases and players. For instance, in [81], ETSI tries to answer the question of how several MEC systems (i.e., a MEC federation) can communicate with each other seamlessly, while in [82] ETSI explores security-related use cases and requirements. Instead, in [83] the standardization body explores the so-called edge native applications (from the "cloud" native) concept and how the ETSI MEC paradigm could support it, while finally in [84] the white paper focuses on communication between ETSI MEC and M2M-IoT devices. At the same time, the MEC concept is being explored and extended by other organizations and researchers. Arora *et al.* [85] propose a new MEC architecture for the Radio Network Information Service (RNIS), based on OpenAirInterface and fully compliant with the new ETSI MEC in NFV standard [77]. This service, present in the MEC platform, allows edge applications to know RAN conditions, to be able to modify their behavior and match the network conditions [86]. They create two different message brokers of the RNIS, one with RabbitMQ and the other with Kafka, with the first one being superior in terms of lightweight CPU utilization. Ksentini and Frangoudis focus on extending ETSI MEC to support LoRa communications while Zanzi *et al.* [87] focus on the introduction of a MEC *Broker* on top of the ETSI MEC architecture, between the OSS block and the tenant (i.e., the UE). The MEC broker enables tenants to access management options such as life cycle management and application administration privileges. In addition, they propose an orchestration solution called M^{2}EC

(from multi-tenancy MEC), which allows for minimizing overall resource utilization. The authors in [88] extend ETSI MEC to support stateful application relocation by leveraging container migration techniques. Castellano *et al.* [89] propose a split MEC architecture, in contrast with the current monolithic ETSI MEC architecture we have described in Section 2.1.1. They argue that standards are not helping the MEC deployment in real scenarios and, at the same time, companies are looking at MEC as an opportunity to save money or generate revenues. Therefore, they propose to further separate the ETSI MEC architecture into Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS) levels, to help the MEC deployment. In [90], the authors propose a constrained MEC (cMEC) architecture to deploy a MEC platform in constrained devices such as terminals or edge robotics). Huang *et al.* [91] present in detail an SDN-based MEC framework, compliant with both ETSI MEC and 3GPP architectures. According to the authors, it provides the required data-plane flexibility and programmability, improving overall latency. Taleb *et al.* [92] propose the concept of a Content Delivery Network (CDN) slice, which is a CDN service instance created upon a content provider's request. They base their proposal on the latest versions of MEC, and NFV and on proposals produced in the frame of 5G standardization efforts. In their work, they tackle QoE-driven cloud resource allocation and elastic resource management. Finally, the authors in [93] propose an open-source framework ETSI MEC and 4G/5G complaint to evaluate the performance of MEC apps, while Cicconetti *et al.* [94] proposed a prototype for a hospital use case, where end-clients and edge apps host Quantum Key Distribution devices, with a complaint ETSI MEC architecture.

2.1.2. Where: MEC flexible provisioning

MEC provisioning is an important feature because, thanks also the degree of flexibility provided by NFV as also shown in the previous subsection in 5G networks, it will help the MEC paradigm to set itself with a primary role in the deployment of future cellular architectures. Efficient provisioning is achieved using both careful MEC resource deployment, and the capacity to follow user mobility. Therefore, this section answers the question *where should MEC resources be deployed?* To answer, we cover two important MEC aspects: (*i*) (flexible) MEC resources deployment and (*ii*) agile migration of MEC resources. Indeed, it is of crucial importance for URLLC applications that devices can reach MEC resources in a few milliseconds or less (i.e., with extremely low latency) and that edge resources are fairly assigned to services [149]. Moreover, the MEC must also support user mobility, which requires rapid service provisioning and fast migration of applications, VNFs, and MEC services. These features enable innovative solutions for a well-known MEC research problem, i.e., for computation offloading. Table 2.1 summarizes the works overviewed in this area.

Flexible MEC resources deployment

Location deployment: A fundamental problem studied in the literature concerns where to physically deploy MEC resources. Some examples can be found in an Intel white paper [117] and several papers in the literature address this topic. In [118], Pérez *et al.* highlight that in future deployments, mobile network operators will have to decide how many MEC *points of presence* are needed, also considering the presence of gNBs. Therefore, they created a model, based on inho-

Table 2.1: List of papers related to MEC flexible deployment

MEC Provisioning	References	Use Case	Tools	Evaluation	Most relevant lessons learned
Computation Offloading	[95], [96], [97] [98], [99], [100] [101], [102], [103] [104], [105], [106] [107], [108], [109] [110] [111], [112] [113], [114], [115] [116]	- IoT - Privacy preserving - Inter task dependency - Energy efficiency - Parallel computing - Autonomous devices - Caching - M2M - Wireless Energy Transfer (WET) - LEO - Digital Twin - RIS	- (Mixed) Integer Program (MIP) - Logic Based Benders Decomposition - Lyapunov optimization - Gibbs Sampling algorithm - Markov Decision Process (MDP) - Deep Learning - Non-convex MIP - Dinkelbach's method - Game theory - DRL	- Numerical simulations	- Adding privacy constraint does not affect (very much) performance (< 5%) - Deep Learning algorithms can save energy up to 87% compared to baselines - Novel 6G scenarios and use cases will increase the complexity of offloading decisions
Flexible MEC resources deployment	[117], [118], [119] [120], [121], [122] [123], [124], [125] [126], [127], [128]	Location deployment	- Model with inhomogeneous Poisson point processes - Voronoi cell-based analysis - Graph-based algorithm - (M)IP - Meta-heuristic	- Simulations with real scenarios - Datasets - Numerical simulations - Prototype implementation - Sumo5G	- Considering the New Radio profiles of 5G, FDD 120 kHz is the one that minimises the number of MEC stations deployment - Up to 50% of traffic can be absorbed by MEC servers - Beyond 5G networks are required to scale up MEC services
	[129], [130], [131] [132], [133], [134] [135], [136], [137] [138], [139], [140] [141], [142], [143]	VNF placement at the edge	- Heuristic - Architecture design - (M)IP - Genetic Algorithm - Randomized Rounding algorithm - Machine Learning	- Numerical simulations - Prototype implementation - Trace-driven simulations - Test-bed	- Both Randomized Rounding and genetic algorithms seem a viable solutions for VNF placement at the edge, with performance close to the optimal
	[144], [19], [145] [146], [147], [148]	Systems deployment	- Framework design - MEC platform deployment - ETSI-MEC with 5G core	- Systems-level evaluation - Open Air Interface prototype - 5G-MEC V2X testbed	- MEC over FiWi could prolong devices battery up to 11.30 h - MEC can reduce latency up to 60% compared to a cloud datacenter - An end-to-end open ETSI-MEC simulator has not been fully developed yet

mogeneous Poisson point processes, which studies the MEC deployment with simulations based on a real topology. Since the cell tower presence constrains MEC deployments, Syamkumar *et al.* [119] analyze a 4M dataset of antennas located in the US to evaluate the MEC deployment in a real case scenario, showing in which areas new network infrastructures are needed. Similarly, in [120], the authors study how to allocate MEC resources as a function of service demand. They propose a graph-based algorithm to provide a partition of MEC clusters, which takes into account the capacity of MEC servers. The authors evaluated it with a mobile communications data set, containing real-world spatio-temporal human dynamics. Furthermore, in [121], the authors study how the mobility of citizens in a city should also affect the optimal placement of MECs. Kherraf *et al.* [122] formulate the problem of MEC resource provisioning and workload assignment for IoT services (RPWA) with a mixed integer programming formulation. Given its complexity, they decouple the problem into two sub-problems: (*i)* delay aware load assignment and *ii)* mobile edge servers dimensioning. Through numerical simulations, they show that their scheme achieves a higher admission rate (from 1% to 44%) compared to the solution proposed in [150].

Miltiades *et al.* [123] provide another way to deploy efficiently MEC resources. Indeed, the authors focus on the control plane, studying the latency of packet transfer and processing inside an NFV environment. To minimize the latency, they design proximity zones around MEC platforms hosting MEC application instances, showing how these zones could help for a flexible and latency-aware use of the MEC platform. Castellano *et al.* [124] propose a distributed algorithm to coordinate the resource allocation in edge computing scenarios. They consider the optimal resource assignment and evaluate its feasibility with a prototype implementation that follows a Pareto-optimal resource assignment. Several recent papers also address this issue, thanks to the increasing interest in MEC and the rollout of 5G networks. For instance, Virdis *et al.* [125] make a performance evaluation, using Sumo5G and CoFLuent, of different MEC deployments in cellular networks (4G, 5G non-standalone, and 5G Standalone) for an infotainment vehicular use case. Chantre *et al.* [126] consider the problem of MEC location with protection schemes, selecting edge locations to place MECs hosting protected slices. Filippou *et al.* [127] also highlight several options for MEC app deployments in 5G networks, highlighting the pros and cons of each deployment. Finally, the authors in [128] study an ETSI-MEC compliant scenario for a MEC app placement scheme in a full federated edge architecture. They formulate an Integer Linear problem and through meta-heuristic Tabu-Search they propose a solution that can instance a MEC application considering constraints such as computing resources, latency, and MEC service availability.

VNF placement at the edge: MEC (resource) placement can be made flexible thanks to the use of VNFs on top of virtualized infrastructure, using virtual machines and containers. Hence, new scenarios are now available to be explored together with the MEC paradigm: the VNF placement and resource migration (which we will overview in the following sub-section). Depending on the type of service, different constraints (e.g., low latency, high compute power, and/or a fixed dedicated uplink/downlink bandwidth) are present and the MEC Orchestrator should be able to decide quickly where to place the VNFs, e.g., near the core of the network or at the edge. In [129], the authors propose a data-driven VNF placement strategy with ONAP across distributed data centers, hence in a MEC scenario. Through simulations, they compare their solution against other ones proposed with an Openstack-based approach, showing that their strategy is better in terms of overhead and data center utilization. Salsano *et al.* [130] propose an architecture for the dynamic deployment of VNFs leveraging on the MEC. According to the principles designed in the Super-

Fluidity project, they decomposed the network functions needed for MEC as software *reusable functional blocks* (RFB), which hence allows for flexibility in the architecture. The proposal has been validated by studying a video streaming service use case.

For the VNF placement problem, many papers propose a mixed or integer linear programming formulation which is then solved by a genetic algorithm. For instance, in [131], the authors build a VNF placement strategy on top of ETSI standards for MEC and NFV MANO. They propose a genetic algorithm, considering access latency and service availability constraints. Through numerical results, they show the feasibility of their algorithm, reaching near-optimal performance. Similarly, Yuan *et al.* [136] propose a genetic algorithm for a latency-optimal placement problem which also considers MEC CPU time, total network latency, and long-term scaling cost. The authors in [137] study a similar problem and solution but in their case they also consider an SDN/NFV-MANO enabled edge architecture, arguing that to obtain better performance it is necessary to consider a coordinated placement of VNFs with SDN, NFV, and MEC. Thiruvasagam *et al.* [138] state that VNFs, MEC and cloud servers, and communication links are subject to failures due to software bugs, misconfiguration, overloading, hardware faults, cyber-attacks, power outage, and natural/man-made disaster and therefore propose a resilient and latency-aware deployment of network slices in MEC cloud facilities problem, formulating it as as a binary integer programming (BIP) model and proposing a genetic heuristic to solve it.

Poularakis *et al.* [132], focus on joint service placement and request routing problem in a MEC multi-cell scenario with multiple constraints, aiming to minimize the load of the centralized cloud. They propose a custom *randomized rounding* algorithm, showing that, in terms of cloud load, they can achieve a 25% better performance with respect to the greedy solution proposed in [151]. Similarly, the authors of [152] explain how to design an edge computing framework, including a service orchestration algorithm. The latter permits to move and place services within 25 ms and it can scale and support services instantiated on a per-user basis. In [133], the authors propose a two-scale framework that jointly optimizes service placement and scheduling of requests under storage, communication, computation, and budget constraints, proving that the problem is NP-hard. Furthermore, they develop a service placement polynomial-time algorithm that reaches performance close to the optimal solution (up to 90%). Moreover, some papers deal with an edge-cloud architecture. Yang *et al.* [134], for instance, study the problem of service chaining with VNFs in a mixed edge-cloud scenario. They minimize the maximum link load ratio under delay constraints. Finally, the authors in [135] study the optimal provisioning of edge services with both shareable and non-shareable resources via joint service placement and request scheduling. They show that the problem is NP-hard and propose several heuristics which are then evaluated via data-driven simulations. The authors in [139] study the problem of VNF placement for IoT applications in edge nodes, proposing an efficient heuristic and validating their results with a test bed. Also, in [140], the authors consider MEC-based VNF placement depending on the type of AR tasks offloaded to MEC nodes, while Behravesh *et al.* [141] propose a mixed inter linear programming problem to solving a joint user association, SFC placement, and resource allocation problem in MEC-enabled 5G networks. The authors in [142] consider a joint problem of communication and computing resource allocation comprising VNF placement and assignment, traffic prioritization, and path selection to minimize the total cost of allocations. To solve their NP-hard problem, they propose a Double Deep Q-learning (DDQL) technique showing its effectiveness using numerical results. Instead, Nemeth *et al.* [143] analyze the problem of VNF placement in a realistic use case-

based scenario (mobile robotics for warehousing solution in the Valencia city haven), showing the need for VNF placement solutions with strict delay bounds and reliability constraints, while taking into account radio coverage, mobility, and battery conditions and proposing an efficient heuristic. Finally, for interested readers, the authors in [153] present a recent state-of-the-art review of VNF placement techniques and also focus on the placement of Containers Network Functions (CNFs) in edge/fog computing, highlighting the most prominent studied techniques while also considering future challenges.

Systems deployment: Some works dig more into system implementation. Rimal *et al.* [144] propose a MEC deployment over Fi-Wi, which is a combination of mmWaves and optical fibers that allows ultra-high speed. The authors discuss the possible benefits of the framework, such as prolonging the discharge of edge device batteries, with a capacity of just 1000 mAh, up to 11.5 hours, depending on the offloaded traffic load. In [19], the authors provide a MEC platform deployment solution for 4G LTE networks using a middlebox, for which they have designed a prototype based on the OpenAirInterface (OAI) cellular platform. Other works propose the integration with different technologies. For instance, in [145], Kempen *et al.* provide the design of the so-called MEC-ConPaas platform, a mobile-edge cloud platform that aims to support future research on edge cloud applications, leveraging Raspberry Pi devices. Their experiments show that it is possible to support real cloud applications with extremely simple edge devices. The authors in [146] show a full-fledged MEC architecture integrated with 5G core (therefore following the ETSI-MEC 5G specification) with an application (video streaming) deployment use-case. Wadatkar *et al.* [147] examined existing 5G-MEC V2X testbeds and categorized them, exposing gaps between the ETSI-defined deployment model and the end solutions and also showing current issues with open source available 5G V2X testbeds. Finally, the authors in [148] proposed an updated survey on some of the systems deployment challenges, highlighting existing testbeds and still open issues in real system deployments such as mobility awareness, offloading decision, AI at the edge, and privacy compliance.

Agile migration of VNF-based MEC resources

In addition, due to user mobility, it is of primary importance to establish a connection between the end user and MEC resources and maintain it throughout all the necessary stages, with the services that should be able to migrate quickly depending on user movements. Table 2.2 provides a summary of works in this area.

Mobility support (Follow me Cloud and Service Replication/Migration): The relation between MEC and the mobility of users and the dynamics of their demands is also an object of investigation. Several works addressed the performance and optimization strategies for migration in a MEC scenario, to keep performance levels high and use resources efficiently.

One particular paradigm developed for user mobility support is the so-called Follow me Cloud, which has been proposed in [154]. Follow me Cloud uses an approach similar to Information Centric Networks (ICN): it proposes the replacement of the IP addressing for a service/data identification. This allows for a continuous connection between mobile user and service, even when service migration occurs. After the initial paper in 2013, Follow me Cloud has been further extended. In [155], the authors compare the Follow me Cloud paradigm with other two testbeds based on

Table 2.2: List of papers related to agile migration of VNF-based MEC resources

MEC Provisioning	References	Use Case	Tools	Evaluation	Most relevant lessons learned
Agile Migration of MEC-VNF-based resources	[154], [155], [156] [157], [158], [159] [160], [161], [55] [162], [163], [164] [165], [56], [166] [167], [160], [168]	Mobility support	- MDP - Integer Linear Programming - LSTM - Lyapunov techniques - DRL	- Numerical simulations - Testbed experiments	- Follow me Cloud with MEC can reduce the iterative migration time up to 61% compared to existing solutions - Increasing the number of service replicas reduces the probability of user reactive migration (from 21% to 26.5%) - Novel ML techniques could help reducing the number of migrations
	[169], [170], [171] [172], [173], [174] [61], [175], [176] [177], [178], [179] [88]	Migration with containers	- MPD - Optimal stopping theory - Architecture design	- Testbed experiments - Data traces simulations - ETSI MEC Prototype implementation - Numerical simulations	- Containers reduce up to 56% the handoff time - A dynamic placement scheduler reduces VNFs migrations up to 94.8% compared to baselines schedulers - Containers achieve from 2x up to 8x faster migration time compared to VMs migration - Different migration strategies should be applied w.r.t. applications features

locator/identifier separation protocol (LISP) and SDN, showing the potential of their paradigm and its feasibility for real-world deployment.

In [156], Addad *et al.* merge the Follow Me Cloud concept with the MEC paradigm, to provide lightweight live migration at the edge, based on container technologies. They evaluate their proposal with a real testbed. According to their results, using Follow me Cloud with MEC would decrease iterative migration time by 50% compared to the baseline solution proposed in [175].

Finally, in [157], the authors focus on the vehicular networking case and develop a new architecture named Follow me edge-Cloud (FMeC). Leveraging on the strict requirement of the automotive vertical, they created an FMeC architecture based on MEC and SDN/OpenFlow principles, and validated their new concept through theoretical analysis and simulation experiments. Instead, Farris *et al.* [158] study the proactive service replication problem, to reduce the overall migration time and guarantee good QoE. They leverage the prediction of user mobility patterns and the overall synchronization of states of service replicas. At the same time, this technique collides with limited edge resources. Therefore, the authors formulate two different optimization problems: one minimizes QoE degradation during handover, while the other minimizes the cost of service replicas. Through simulations, they show that increasing the number of service replicas would reduce the probability of user reactive migration by up to 26.5%. Similarly, the authors of [159] deal with the fast relocation problem of services due to user mobility, investigating container-based virtualization techniques. In their work, they support the use of mobility in a MEC infrastructure by designing a framework with three different modules (*Service Manager* for monitoring applica-

tions, *Edge Manager* for container placement, and *Edge Orchestrator*, which manages the overall framework) to guarantee fast response time and exploiting service replication. They show the benefits of classic migration procedures. They further state that this framework may also be integrated into the ETSI MEC architecture.

Several recent papers consider other proactive techniques for service migration in a MEC scenario. The authors in [161] show the benefit of using a LSTM network to predict MEC services deployments. In particular, their algorithm has to choose between four deployment actions according to user mobility: migration, replication, scale, and retain. Labriji *et al.* [55] study an IoV MEC scenario and propose a novel proactive technique (considering both vehicles' trajectories and multi-modal mobility estimation) for the online migration of computation service in vehicular 5G networks. Shah *et al.* [162] propose to use SDN to provide a unified control plane interface in a scenario with several MEC nodes to perform effective network and service mobility management for vehicles (i.e., allowing seamless service migrations) while managing also the resource-constrained MEC servers. The authors in [163] use Lyapunov techniques to decide the optimal time to trigger service migration and select which MEC Hosts are better placed to host the migration of the running service. The authors in [164] focus instead on the minimization of migration events rather than on maximizing the usage of resources. They state that service migrations can create significant service downtime to applications that need low latency and high reliability, while also in addition to increasing traffic congestion in the underlying network. They propose an induced service migration minimization (PrISMM) algorithm after defining a Markov Decision Process and show the effectiveness of their solution by numerical results. Liang *et al.* [165] state that offloaded tasks can be seamlessly migrated between base stations without compromising the resource-utilization efficiency and link reliability and therefore they propose a a policy for migration/handover between base stations by jointly managing computation and radio resources, developing an efficient relaxation-and-rounding based solution. Other studies leverage machine learning techniques to solve the problem of mobility support. For instance, both [56] and [166] use (deep) reinforcement learning for their task/service migration problem supporting end-users mobility in a cellular network with MEC nodes, while in [167] the authors consider a deep learning approach to allocate and migrate UE requests to edge servers. Finally, a few papers consider also privacy aspects for users mobility. Indeed, Sangaiah *et al.* [160] propose to leverage machine learning techniques on MEC nodes to preserve position confidentiality of roaming users, arguing that MEC servers would help maintain both a low latency service and position confidentiality. The authors in [168] formulate the service migration progress as a joint optimization problem that minimizes service response latency due to service migration and location privacy leakage risk, ensuring protecting user's location privacy.

Migration with containers: Resource migration mainly deals with VNFs, e.g., migration of the virtual machines and containers that run the VNFs, across different hosts. Normally, in a centralized data center most of the virtualized resources and migrations are for virtual machines. However, placing resources at the edge of the network leads to the deployment of small data centers in which it is not possible to execute the same virtualization technologies of typical large data centers [169]. Therefore, services would be better deployed using containers, which represent a lightweight solution for deployment and migration. Therefore, many studies focused on container migration. The authors of [170] evaluate Docker, which is the most commonly adopted and powerful container technology as of today, at least in the scenario of edge computing. They base their

evaluation on four different aspects: deployment and termination, resource and service management, fault tolerance, and caching. They show that Docker is a valid candidate platform for edge computing. Furthermore, Avino *et al.* [171] state that a key beneficial feature of MEC would be the ability to ensure server portability with low overhead. They show that this can be achieved using Docker. To prove this, they quantify Docker CPU utilization in two use cases in an experimental setup: online gaming and video streaming. In both cases, the Docker overhead was quite small, even though for the online gaming case the overhead slightly increases with the number of supported servers. Wang *et al.* [172] state that in a MEC scenario, the migration of resources is difficult to perform since the environment is very dynamic and volatile. Hence, they propose a Markov decision process to deal with this uncertain scenario, validating their model using mobility traces for San Francisco taxis. Recently, Doan *et al.* [173] have proposed a measurement framework to study the existing data center migration approaches in a MEC scenario. They show that these approaches are unfeasible due to the high migration time, causing therefore substantial service degradation.

The papers mentioned above do not consider stateful migration. With stateful migration, the service is migrated and resumed in the exact state in which it was before migration, without losing connection with the users. In [174], the author's goal is to achieve a seamless live migration, with a focus on reducing the file transfer size during the migration procedure. They study Docker layered storage and propose to share common storage layers across Docker hosts to reduce file transfer size. They propose and evaluate a prototype, which shows interesting performance improvements (up to 56% reduction of hand-off time with respect to reference approaches defined in [180]). In [61], the authors argue that containers would be fundamental for meeting low latency requirements. They study state-of-the-art migration techniques with Docker and with virtual machines using KVM. Moreover, they propose an application-level live migration protocol that eliminates common drawbacks like the lack of hardware abstraction at the host. The work of Machen *et al.* [175] proposes a 3-layer framework for supporting stateful live service migration encapsulated in containers in a MEC scenario, to ease the implementation with popular container and virtual machine technologies. They validate their solution with small-scale experimental results, showing that containers can achieve from 2x up to 8x (depending on the scenario evaluated) faster migration times compared to VMs migration. Finally, Cziva *et al.* [176] propose a more general framework for VNF migration. They focus on a dynamic placement of VNFs at the edge of the network and especially on the dynamic re-schedule of VNF placement. Their approach leverages optimal stopping theory. They run simulations based on a nationwide backbone network with real-world ISP latency and show that their solution incurs much fewer VNF migrations (up to 94.8%) than other existing migration schemes.

Other recent papers also considered container migration in a MEC scenario. Hathibelagal *et al.* [177] assess the viability of different migration strategies for three different containerized MEC applications (for V2X use case), using a real testbed and the ETSI MEC Sandbox. They show that depending on the features of the application, a different migration strategy should be applied. Some of the same authors in [178] propose a testbed with Kubernetes to support the migration of MEC apps between two MEC Hosts. The authors in [179] propose ShareOn, a framework with dynamic container migration, validated using a set of edge-cloud nodes distributed in San Francisco and which aims to execute a real-time application to detect license number plates in automobiles. Finally, Barbarulo *et al.* [88] extends ETSI MEC to let it support stateful application relocation by

running applications as containers and exploiting existing container migration technologies.

Computation Offloading

A well-known research problem coupled with MEC is the computation offloading problem. Indeed, thanks to the deployment of edge resources and their agile migration, it is possible to offload computation tasks from mobile users, with benefits for instance on the device battery life. Computation, according to [181] could be *fully* offloaded to the MEC, *partial* offloaded, or utterly processed at mobile device (*local execution*). Hence, this new paradigm raises new questions and challenges in the MEC resources deployment domain, such as the trade-off between minimizing device energy consumption and achieving acceptable execution delay due to offloading. Of course, the delay also depends on the MEC resource deployment [181]. Another problem is in identifying the edge server that should be selected for offloading. In [95], the authors propose a delay-sensitive IoT services scenario, in which task offloading is jointly considered with (MEC) resource allocation and (task) scheduling. They formulate the mixed-integer problem "Dynamic Task Offloading and Scheduling (DTOS)". Due to its complexity, they decompose the problem using a technique called *logic-based benders decomposition* and perform several simulations to check the effectiveness of their proposed solution. Finally, with the same algorithm, they evaluate trends in different vertical industries, namely tactile Internet, Telesurgery, Factory Automation, ITS, and Smart Grid, with variable latency requirements. In [96], the authors focus on the privacy aspect of offloading to a MEC server. They show that existent privacy-preserving techniques do not work well in this new edge scenario and so they create PEACE, a scheme that jointly considers privacy-preserving and cost-efficient task offloading. According to their experiments, adding the privacy constraint does not affect very much the overall performance ($\approx 5\%$). Yan *et al.* [97] study the inter-user task dependency in an MEC system. First, they focus on a two-user MEC scenario (in which the input task of a user requires the output task of the other user). Their goal is to minimize both the energy consumption of users and task execution time through an optimal task offloading policy and resource allocation problem; the problem is further solved using a reduced complexity Gibbs sampling algorithm. Further, they extend the scenario to a general multi-user MEC, in which an input task of a user requires final task outputs from multiple users. They evaluate this extension with the same algorithm proposed for the single-user case and find that their solution performs well compared to other sub-optimal schemes. Meng *et al.* [98] aims to achieve a delay-optimal computation offloading policy for computation-constrained MEC systems, taking into account also the future delay performance of the MEC system. To deal with this problem, they create a finite horizon Markov decision process (MDP) for two cases: single-user single-MEC server and multi-user multi-MEC server scenario. They manage to derive a closed-form multi-level water-filling computation offloading solution and show via simulation that it outperforms other schemes proposed in [182] and [183] by $\approx 4\%$ in terms of average delay. In [99], the author's goal is to improve the energy efficiency of a MEC system hosting both URLLC and delay-tolerant services. To solve this problem, they use a Deep Neural Network (DNN), trained with a so-called digital twin model (a virtual digital model that merges data from the real network and fundamental rules from theoretical studies), showing the benefits of their DNN framework. Compared to baselines, it enables energy savings of up to 87%. In [100], the authors state that virtualization on shared I/O resources, which could happen in an edge computing scenario, might lead to computation degrad-

ing (meaning that the speed of VMs sharing the same hardware might degrade due to interference). Therefore, they study the problem of joint radio-and-computation resource allocation (RCRA) in multiuser MEC systems in the presence of I/O interference, showing that their solution performs well against optimal algorithms ($\approx 4\%$ of difference). In [101], Josilo *et al.* focus on the coordination problem of offloading to the MEC decisions of autonomous devices, such as vehicles, drones, or manufacturing machines, to minimize device energy consumption and task completion time through a game theoretical analysis. Wang *et al.*, in [102], portrays a joint optimization problem on the computation offloading and content caching strategies for wireless cellular networks with MEC. They propose an alternating direction method of multipliers (ADMM) algorithm, evaluating its effectiveness with different system parameters.

In [103], the authors focus on the integration between virtualized Small Cell Networks (SCNs) with MEC. Their solution might help in reducing the energy consumption of UEs thanks to offloading procedures. However, complexity might explode. The authors formulate the problem as a mixed integer nonlinear program and then transform it into a biconvex problem. Through simulations, they compare it against the optimal and an algorithm proposed in [184]. Their solution achieves better performance, with a gain of about 20% against [184], while nearly reaching optimal performance.

The authors of [104] state that nowadays Machine-to-Machine (M2M) communications attract ever-growing attention. Differently from other communications networks, M2M uses high-frequency small packet size, therefore needing a special optimization of both energy consumption and computation. Therefore, the authors introduce a MEC architecture for virtualized cellular networks with M2M communications, to decrease energy consumption and optimize the computing resource allocation. They create an observable MDP to minimize the system cost. Mao *et al.* [105] propose to use Wireless Energy Transfer (WEF) to prolong device battery life. However, it is hard, for the MEC system, to jointly schedule radio and computational resources as well as energy utilization maintaining at the same time the overall performance requirements. Hence, they study energy efficiency and delay in a multi-user wireless powered MEC system with multiple access schemes. They design a low-complexity online algorithm based on Lyapunov optimization theory, allowing them to transform their problem into a series of deterministic optimization problems. Through theoretical analysis, they show that their algorithm allows to trade off energy efficiency for delay.

Sardellitti *et al.* [106] formulate the computation offloading problem, from the mobile users to the cloud server, in a multi-cell mobile edge computing scenario. They define it as the joint optimization of radio and computational resources to minimize multi-user energy consumption under latency constraints. They find that in the MEC scenario, offloading becomes more convenient with high computational loads.

Thanks to advanced techniques, novel technologies, and future 6G scenarios, different novel approaches have been considered in the study of computation offloading at the edge. For instance, in [107] the authors study the problem of computation offloading and resource allocation in a Digital Twin MEC scenario using Federated Learning techniques. Others instead use distributed deep reinforcement learning [108] or Multi-Agent deep reinforcement learning [109] to the “standard" deep reinforcement learning. Novel technologies also give new scenarios to the computation offloading problem, such as Reconfigurable Intelligent Surfaces (RIS) coupled with MEC [185],

Low Earth Orbit (LEO) satellites [110], [111], O-RAN [112] and 6G scenarios [113]–[116] Finally, for interested readers, different surveys give a systematic review of machine learning for computation offloading [186], [187].

Summary, lessons learned and future research

We now summarize lessons learned from the existing work and highlight potential improvements to existing solutions. We also identify possible future research directions.

MEC deployment: One of the most important novel aspects of MEC is its proximity to the UE. This leads to new unexplored scenarios and gives the possibility to enhance different features, such as computation offloading. However, MEC provisioning is challenging. Indeed, network providers should carefully consider both the QoS required by services (e.g., the ones using URLLC) and the cost of deploying and maintaining a new edge infrastructure. In [117], to achieve this tradeoff, the authors propose to expand the existing infrastructure, i.e., the network provider's towers and offices. Otherwise, new deployment possibilities lay inside the *last mile* network, thus helping with the development of the *smart city* paradigm. Hence, new sites could be stadiums, private/public buildings, enterprises, or homes. Moreover, the NFV paradigm introduces a degree of flexibility. For instance, it will be possible to create a disaggregated MEC architecture, where the MEC Orchestrator is placed in a more centralized node and MEC hosts are instead more decentralized, nearer to users. Thanks to NFV, MEC can also be placed in different parts of the cellular networks, as shown in [125]. At the same time, NFV allows MEC to support user mobility by migrating virtual resources across the edge infrastructure.

From the papers surveyed, the key **lessons learned** are:

- Computation offloading is one of the most studied paradigms within MEC and, in general, edge computing. Hence, the following considerations account only for the most recent papers in this area. Most of the papers focus on minimizing device energy consumption playing with offloading tasks, resource allocation or delays ([95], [97]–[101], [103]–[106]). Among all results, we would like to mention the performance of deep learning which, according to [99], can save devices energy up to 87%. Moreover, the authors of [96] consider also privacy issues, showing that the privacy constraint does not affect performance very much (i.e., within $\approx 5\%$). However, it would be interesting to see more works considering new scenarios such as user mobility within different MEC hosts and 5G features such as network slicing [71]. During the writing of this thesis, we found only two works that consider slices into a MEC scenario [126], [138]. Novel application scenarios, e.g., cloud gaming or Digital Twin, are not fully explored. Similarly, there is a need to study VNF (applications) sharing (like in [188]) or the revenues/economical costs of offloading decisions, possibly also leveraging both novel artificial intelligence and computing tools. More recently, there have been several papers that consider computation offloading with MEC in several next-gen scenarios, such as LEO [110], [111], UAV, and advanced ML techniques [108]. For interested readers, we mention the survey of Mach *et al.* [181] on MEC with computation offloading.

- Several papers focus on the MEC location deployment. While [120] confirms the benefit

stemming from the presence of MEC, other authors provided useful insights on possible MEC deployments according to 5G constraints [118], [119], [189]. They consider smart cities, as well as industrial and rural scenarios. The authors of [121] study the MEC deployment in a smart city, considering pedestrian mobility. The authors of [123] enhance MEC host deployment by designing proximity zones around MEC platforms, helping them to become more latency-aware. Finally, the authors of [122] and [124] provide a more theoretical approach. [124] considers a decentralized orchestration while [122] shows that in terms of admission rate their scheme reaches higher performance (from 1% to 44%) compared to [150]. Thanks to the ongoing rollout of 5G networks, new papers focus on the MEC deployment problem, studying scenarios and proposing different solutions that are also ETSI MEC compliant [125], [127], [128].

- Afterwards, different papers focused on VNF placement at the edge. Several approaches have been proposed, ranging from different frameworks [129], [130], [152] to theoretical works [132]–[135], [141], [142]. The latter mostly focused on formulating mixed-integers programming problems, with genetic algorithms proposed as the main solution in different works [131], [136]–[138].

- Finally, some systems-related works have been highlighted [19], [144], [145], giving several interesting insights. For instance, MEC reduces latency up to 60% compared to a cloud datacenter. More recent works also considers or propose ETSI-MEC compliant testbeds [146]–[148]. More specifically, for interested readers the authors in [147], [148] provide an updated state-of-the-art view of MEC test-beds and tools.

Concerning possible future work on MEC deployment, we mention the following points:

- It would be interesting to investigate more location deployments and VNF placement resources at the edge, especially in real-case system-oriented scenarios, leveraging new possibilities given by novel network protocols, standardized interfaces [147], technologies such as UAV and LEO ([110], [190], [191]) and new scenarios such as Digital Twin for 6G networks [115], [116]. ML techniques could also help in this particular scenario. For instance, the authors in [192] use a Graph Neural Network (GNN) model to plan the location of novel 5G cells. Something similar could also be proposed for edge node deployments, also keeping in mind issues such as network scalability and constraints such as QoS, QoE, CAPEX, and OPEX.

- Future works on MEC deployments should also consider standardization efforts made by SDOs such as ETSI MEC and O-RAN, also pointing out possible shortcomings and filling these missing gaps (some examples are [85], [72] and [193]). During the draft of this thesis, we noticed that while ETSI MEC-compliant solutions are becoming more and more popular, there are still some open challenges, as also stated in [147].

- Furthermore, researchers should exploit new scenarios as the ones proposed at the end of this section, under *other research challenges*.

MEC migration: Regarding the agile migration of VNF-based MEC resources, two main paths have been evaluated:

- **Mobility support** given by the stateful Follow me Cloud paradigm [154], [155], which is shown to work better than solutions based on LISP and SDN. The authors of [156] and [157] have merged that paradigm with vehicular networks and MEC, whereas [158] and [159] propose a proactive service replication to reduce migration time. They showed that increasing the number of service replicas would reduce the probability of user reactive migration (from 21% to 26.5%). Recent papers also use ML techniques. For instance [160] proposes to leverage both MEC and ML for maintaining services position confidentiality, while the authors in [56], [161], [166], [167] use deep reinforcement learning or deep learning (e.g., LSTM) solutions showing their feasibility in supporting users mobility at the edge. Another possible direction could be to use SDN [162] to provide a unified control plane interface to perform effective network and service mobility management, especially in an IoV scenario.
- **Migration** of VNFs and especially containers, since the latter is more lightweight to VMs (an important feature in a scarce-resource edge infrastructure). Indeed, according to [175], containers achieve from 2x up to 8x (depending on the scenario evaluated) faster migration times compared to VMs migration. This thesis analyzed preliminary stateless migration papers ([170], [171], [173]) and more recent works, focusing on stateful (live) migrations ([61], [174], [175]). We have also addressed some more theoretical works ([172], [176]). Recent papers provide also novel test-beds and frameworks to study container migrations for the edge scenario [88], [177]–[179].

Future works on migration should consider the following points:

- There is a need for evaluating the performance of network protocols (e.g., segment routing per IPv6 [194]) for the migration and connection of MEC resources, thanks to their ability to support Service Function Chaining. Another interesting paradigm worth studying in this scenario is Intent-Based Networking.
- While VM migration has been deemed too heavy and slow for an edge infrastructure, more work is needed for understanding stateful lightweight migration, also considering new paradigms such as serverless computing [195], which seems the most promising feature to guarantee smooth QoE.
- New works should also consider new scenarios mentioned in the following paragraph.

Other research challenges. Finally, some possible **open research challenges** can be identified:

- **Privacy** and **security** are still open challenges for MEC [196]. In the design of the new generation of network infrastructure, privacy, the protection of data in general, and security are becoming important new constraints to consider. So far, the literature provides only a limited overview ([96], [126], [160], [197], [198]), while many subjects (authentication between edge/core, proper encryption, how to provide access only to secure devices, etc.) remain unexplored. Moreover, security attacks can also happen during VM migrations, in compromised VNFs (which might be migrated and accessed in another location with fewer

security policies) but also with physical hardware (power cutout or NFV state Manipulation Attack [199]). Researchers should consider all these threats when considering resource migration or deployment. For interested readers, the survey of Khan *et al.* [199] outlines several security and privacy threats in 5G and NFV systems, commenting on potential solutions.

- In recent years, the use of **Artificial Intelligence** at the edge has gained traction, thanks to novel techniques and powerful computing hardware to train ML models. Many works in the literature use (deep) reinforcement learning and deep learning techniques for the allocation and migration of resources at the edge. Others also leverage federated learning [200] or distributed learning [201], which also have the benefit of considering privacy and security issues (since the training is performed locally to where the data is located). Thanks to the AI wave, many new techniques and models are being released and therefore researchers should consider applying them in their work (Graph Neural Networks, Generative Adversarial Networks to name a few) since they promise dramatic improvements in terms of efficiency and cost reduction, especially for use cases involving complex systems and cyber-physical systems. However, one of the main problems of using deep learning techniques is that they are a kind of “black-box" and therefore it is hard to understand how they operate and make certain decisions. Therefore, it is also important to consider Explainable AI and Generative AI techniques in the edge scenario, to understand how and why the machine learning models apply certain decisions. This would be beneficial not only for researchers but also for network operators, who could then implement those techniques in their systems.

- While many papers try to minimize the energy consumption at the user side, it is still unclear how to minimize the energy consumption at MEC side, exploiting therefore the **green MEC** paradigm [202]. Already nowadays, data centers are one of the most energy-consuming infrastructures, and the deployment of new resources at the network edge will surely increase energy consumption, together with capital expenditures of network providers. Indeed, one of the main goals of future 6G networks is to create a sustainable infrastructure. Exploiting hence green energy at the edge (wind, photo-voltaic, etc.) represents a possible solution to overcome these issues [203] [204]. On the line, researchers should also consider the energy efficiency of their solutions and/or architectures. For instance, it is known that ML is a powerful tool but also that the training ofML models is energy-expensive. Using more advanced and complicated techniques would also mean using more energy, with the possibility of raising the carbon footprint of the network infrastructures. Some possible solutions could be to leverage the presence of renewable energy at the edge, or the use of specialized hardware for training (e.g., GPU).

- More system-oriented literature, considering hence standards or novel network protocols and infrastructures, would help in understanding the suitability of MEC in real-case scenarios in the wild. For instance, it would be interesting to evaluate the integration of MEC with several open-source projects and new Internet architectures such as hybrid ICN (hICN), Recursive Inter Network Architecture (RINA), Intent-Based Networking and programmable network tools such as Open Flow or Net-FPGA [205]. The authors in [147], [148] provide an updated state-of-the-art of MEC test-beds and tools but also highlight pending issues and shortcomings.

- Finally, while 5G is slowly becoming widely available thanks to its ongoing rollout, there are already preliminary research efforts on 6G. For instance, as we mentioned before, sustainability will play a major role in the future cellular network but the advancement of hardware and virtualization will also open the door to novel scenarios. For instance, in the so-called **edge-cloud continuum**, different MEC applications and use cases may require varying levels of centralization or distribution (edge/fog/cloud). Also, the growing presence of heterogeneous computing models that exploit hardware acceleration solutions, including the likes of FPGAs, GPUs, or ASICs, could create novel challenges in supporting specific applications. Other research directions could be the creation of a **Quantum MEC**, investigating the potential impact of quantum computing on MEC architectures and the integration of blockchain to enhance the security, transparency, and trustworthiness of MEC systems.

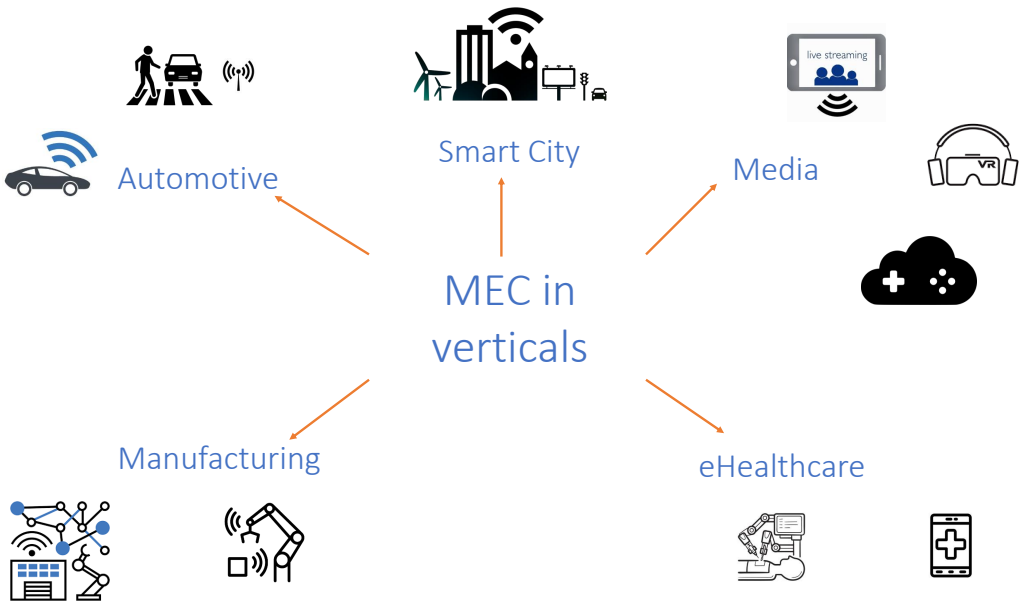


Figure 2.6: MEC in industrial verticals.

2.1.3. For what: Verticals industry

Future network infrastructures will be able to serve different verticals simultaneously. Nevertheless, verticals require different constraints, which can be handled through the MEC. For instance, MEC will guarantee low-latency computing resources with a high degree of flexibility offered to verticals and network providers. Here, we explore the impact of MEC on the most relevant vertical industries defined by 5G-PPP[5], as also summarized in Fig. 2.6: Automotive, Smart City, Media, eHealthcare, and Manufacture. In particular, we will give an updated review with respect to our survey [1] and later comment on how the literature advanced in these three years. Table 2.3 and Table 2.4 summarize surveyed papers focusing on vertical industries.

[5]https://5g-ppp.eu/verticals/

Table 2.3: List of papers related to automotive and smart city verticals

Industrial verticals	References	Use Case	Tools	Evaluation	Most relevant lessons learned
Automotive	[206], [207], [208] [209], [210], [211] [212], [213], [157] [214], [215], [216] [217], [218], [219] [220], [221], [222] [223], [224], [225] [226], [227], [228] [229], [230],	- Safety - Avoid collisions between vehicles and vehicles/ Vulnerable Road User - Advanced driving assistance (ADAS) - Computation offloading - Vehicular clouds - Infotainment	- Optimization Models - Fuzzy logic algorithm - Collision avoidance algorithm - Network model - Graph theory - Stackelberg game - Age of information - DRL	- Numerical simulations - Network simulations - Real testbeds	- MEC hosts reduce latency up to 80% compared to common network architecture - Integrating several access technologies together (sub-6 GHz band, mmWave and IEEE 802.11p) improve performance in highly dense scenarios with low bandwidth - Autonomous cars detect 100% of collisions. With human drivers, the number slightly decrease by 14% - Using Deep Learning for infotainment caching reduces backhaul traffic by 61% - Edge servers cooperation could prevent dangerous collisions during users' handover - Edge computing could help supporting several ADASs scenarios
Smart City	[193], [231], [232] [233], [234], [235] [236], [237], [238] [239], [240], [241] [242], [243], [244] [245], [246], [247] [248], [249], [250] [251], [252], [253] [254], [255], [256] [257]	- Augment ETSI MEC standard to support Smart City - Smart Home - IoT-Based energy management in smart cities - Access service rate for big crowds - Task scheduling for smart city applications - Blockchain for sharing economy services - Security threats on physical layer - privacy preserving	- Big Data - DRL - Optimization - Lagrangian function - Deep Learning - Lyapunov theory - Data model (Ontology) - Federated Learning - Blockchain	- Numerical simulations - Testbed simulations - Trace-driven simulations	- Cooperative DRL (leveraging both cloud-edge resources) reduces delay up 25% and energy cost up to 60%, compared to an only-cloud based solution - DRL can also achieve higher service access rate for big crowds compared to baseline solutions (OSPF and EOSPF) - Physical layer security can be added in a heterogenous IoT scenario thanks to it low complexity and resource allocation - UAVs could support some Smart Cities use cases - The use of Blockchain could increase security and privacy

Automotive

From the early 1990s, Intelligent Transportation Systems (ITS) have been studied to exploit communications between vehicles and infrastructures, to improve the safety and efficiency of transportation. In this context, IEEE developed a new communication protocol called IEEE 802.11p [258]. However, that standard presents several limitations such as poor scalability and lack of performance guarantees. One way to help meet the tight requirements of automotive systems is to leverage cellular networks, e.g., using the C-V2X (Cellular-based vehicle-to-everything) communications paradigm, first proposed for LTE and now extended to 5G networks. Especially 5G should become one of the most important enablers for vehicle communications since, thanks to SDN, NFV, and MEC technologies, it aims to achieve high reliability jointly with low latency (i.e., with URLLC-based slices) [259].

Several papers pointed out the benefits for the automotive industry from leveraging MEC systems. In [260], the authors explain the motivations behind using MEC in ITS, stating that IEEE 802.11p and pure cellular networks might not be sufficient to serve the stringent requirements of the automotive industry. Instead, using the MEC can guarantee reliable low latency communications, seamless service delivery, and highly localized computing resources, necessary to achieve effective C-V2X connections, as further confirmed using extensive simulations in [261]. The authors of [260] also outline possible research challenges such as resiliency, security and privacy, resource management and orchestration, and cooperative awareness among others.

However, in literature, most of the works focus on technical challenges such as enabling edge communications with different access technologies, computation offloading, ad hoc computing resources (vehicular clouds), or supporting driving paradigms such as platooning.

For instance, Hu *et al.* [213] propose a MEC framework for automotive systems, composed of different communications technologies (mmWave, IEEE 802.11p and licensed sub-6 GHz band), to supply services and contents to vehicles. They show through simulations that the adoption of three different access technologies outperforms solutions with only mmWave or sub-6 GHz + mmWave access technologies in various scenarios, especially in the highly dense and low bandwidth ones.

Furthermore, in 2017, the 5G Automotive Association (5GAA) defined the concept of Cooperative Intelligent Transportation Systems (C-ITS), stating that edge computing and in particular MEC will be the enabling technology for V2X communications. In their white paper [262], 5GAA proposes to categorize the main use cases into four groups (as the ETSI MEC standard for V2X does [80]):

- ***Safety:*** This group studies how to avoid collisions between vehicles, for instance at an intersection.
- ***Convenience:*** This group provides time-saving services to manage data and the health of the vehicle (such as the delivery and management of automotive software updates).
- ***Advanced Driving Assistance:*** It includes cases such as traffic signal timing. improving traffic flow, Real-Time Situational Awareness, Cooperative Lane Change (CLC) of Automate Vehicles, and High Definition Maps. According to [262], for its processing of a large

amount of data with low latency and high reliability, this is the most challenging use case for MEC.

- ***Vulnerable Road User (VRU):*** Finally, this group studies communications between vehicles and pedestrians.

Safety and VRU. In the literature, several papers focus on safety issues or VRU discovery with MEC. For instance, Nyuyen *et al.* [208] discuss a method to avoid a collision between pedestrians and vehicles, deploying a MEC server near a base station. This deployment would help smartphones to save energy, giving the possibility to offload the calculation of the Collision Detection Algorithm (CDA) to the MEC server and therefore avoiding both the smartphone battery drainage and calculation latency issues. Through simulations, they show that this solution would improve phone energy efficiency. In [206], the authors propose an enhanced collision avoidance (eCA) mechanism, placed in a MEC server, based on both a Collision Avoidance Algorithm (CAA), and a Collision Avoidance Strategy (CAS). The first algorithm evaluates future vehicle trajectories through beacons while the second strategy decides which vehicles should slow down to avoid collisions. They perform simulations based on SUMO and NS-3, showing the benefits of their strategy by reaching almost 100% of avoided collision in all the scenarios evaluated. Malinverno *et al.* [209] extend the collision detection algorithm, showed in [210], to avoid collision between pedestrians and vehicles, leveraging a MEC-based architecture. Through a detailed simulation scenario, they showed that with autonomous cars 100% of the collisions can be detected on time before the accident happens, while with human drivers, the number decreases by 14%. In [207], Avino *et al.* developed a MEC platform based on ETSI standards and OpenAirInterface, to support automotive systems with tight latency requirements such as safety services. In their simulations, they show that it is possible to obtain better performance in terms of end-to-end delay to the cloud-based approaches ($\approx$ 25%-30%).

More recent papers also considered the impact of edge computing resources in the VRU use case. For instance, the authors in [224] study the problem of collision risks between users associated with different edge servers. Indeed, users located at the boundary of one edge server domain could receive late or miss the alert entirely, putting them in a dangerous situation. Therefore, they propose an edge server cooperation mechanism to prevent users from receiving alerts when they are in the middle of changing the edge servers they are anchored to. Teixeira *et al.* [225] propose a multi-sensing and communication algorithm to prevent potential accidents between vehicles and VRUs. Indeed, to predict accidents, they leverage information from smart city sensors, OBUs, in the VRUs (e.g., smartphones and smartwatches), and on the road itself (e.g., video cameras, radars, lidars). They test their solution in a real environment with real infrastructure, showing that it achieves small latencies, high accuracy, and scalability. On the same line, the authors in [226] realized a multi-operator MEC live trial with a VRU use case. In their paper [227], the authors illustrate and evaluate the impact of using C-V2X sidelink and 5G-NR radio technologies to prevent critical pre-crash situations while Barmpounakis *et al.* [228] propose a novel V2X service and algorithm, namely VRU-safe, capable of identifying and predicting potential imminent road collisions between vehicles and VRUs. They also highlight how the pervasive deployment of edge servers (therefore managing a higher number of OBUs) could lead to better results. Instead, Emara *et al.* [229] use Age of Information (AoI) to measure the impact of the packet inter-arrival time on the timeliness of VRU messages arriving at nearby vehicles, while in [230] the authors

propose a mobility-aware workload orchestration model for VRU safety applications. For interested readers, an updated survey on vehicle-to-pedestrian (V2P) communication considering VRU has been published [263].

Advanced driving: platooning. A key ITS application that will benefit from the presence of MEC is platooning [264]. The latter is a paradigm that allows a group of vehicles to drive together, in line, decreasing the distance between vehicles. This allows for an increase in the number of vehicles on the roads without incurring traffic jams, to augment safety, and to save money on fuel, thanks to the drag effect, thus limiting the overall emissions. Platooning requires very low latency because vehicles can travel several meters in a fraction of a second and a fast access to computing capabilities. Figure 2.3 shows an example of platooning leveraging a possible MEC deployment whose applications to platooning have been recently proposed in a few works. As an example of system design, Montanaro *et al.* [219] present a 3-tier architecture for controlling and managing platoons of vehicles using cloud and edge computing capabilities. Furthermore, the authors of [220] propose a MEC architecture to avoid shock waves, for instance, due to asynchronous brakes, during platoon driving. As an example of computing opportunities offered by platooning, the authors of [265] study the offloading decision of collaborative task execution between a platoon and a MEC server, to minimize task offloading decisions. In [221], the authors provide a framework where the MEC performs a platoon formation and coordination algorithm, receiving periodic updates from vehicles on speed and position. They show that their algorithm achieves low computations and delays in a realistic LTE-Advanced simulation scenario. Quadri *et al.* [266], [267] state that a MEC centralized control of speed and acceleration of platoon vehicles is a viable alternative to common distributed approaches such as V2V communications. Through a detailed Python simulator, they show how, notwithstanding the impact of delay and packet loss probability caused by new UL/DL communications towards the RAN, a MEC centralized control of platoons in 5G networks will help in reducing fuels costs (i.e., allowing smaller inter-vehicle distances), while at the same time supporting a large density of vehicles without incurring in congestions.

Recent papers also study the platooning use case. For instance, in [268], the authors propose a high-level ETSI-aligned architecture for MEC-assisted platooning control, where the centralized platoon controller is a virtualized application running on an edge server, showing that it could be a promising solution for support platoons with all driving-related tasks (e.g., joining a platoon, leaving the platoon). Nardini *et al.* [269] propose the paradigm of Platooning-as-a-Service (PlaaS) in a multi-operator ETSI MEC environment. In their paper, they describe a comprehensive software architecture to implement ETSI MEC-based platooning in a multi-operator environment. In the latter, operators leverage MEC federation to share, in a controlled way, the information that allows users to locate, join, cruise along with, and leave, platoons formed by vehicles subscribed to multiple operators. Carletti *et al.* [270] state that redundant context information from nearby vehicles in the platoon can increase computational costs for the Platoon Leader and that a possible solution could be to form vehicular micro-clouds to enable collective data processing and aggregation, thus reducing the Platoon Leader's perception workload. Their solution is called Platoon Local Dynamic Map (P-LDM) and the idea behind it is to create a single database of context information, distributing the data aggregation load among all members of the platoon. Other papers focus also on computation offloading in a platooning scenario. The authors in [271], [272] study how UAV-assisted MEC can support a platoon of vehicles while Zheng *et al.* [273] consider the general problem of optimizing the allocation of both the communication and computing resources for

vehicles offloading their computing tasks to other platoon members. Finally, the authors in [274] use deep reinforcement learning to train vehicles to form platoons when sharing a similar path to decrease the amount of fuel consumption.

Other advanced driving assistance: While platooning has been extensively studied by researchers, also other advanced driving assistance use cases have been investigated in recent years. Liu *et al.* [275] want to achieve real-time global information in high-definition maps and therefore they propose to share perception information among connected and automated vehicles. To achieve this goal, they design both a data plane to detect, match, and track objects on the road and a control plane with two new algorithms to schedule vehicles and optimize offloading decisions under network dynamics. Tesei *et al.* [276] propose an architecture for deployment at the edge of real-time and mission-critical autonomous driving applications. In particular, they describe and show how this particular architecture could support a Cooperative Autonomous Driving Maneuver Control application for the cooperative lane change use case. Focusing on the architecture side, in [277] the authors propose to merge ICN and NFV in 5G networks to support Advanced Driver-Assistance Systems (ADASs) based on AR. This novel architecture has the double goal of both high mobility and real-time requirements of ADASs and resource orchestration and service management of big data in intelligent transportation systems and they will implement it in a real scenario. Similarly, Giannone *et al.* [278] focus on a scenario with the double goal of supporting the QoE of an in-vehicle infotainment video delivery service, while taking into account the required bandwidth for coexisting high-priority services, such as ADASs. They build their approach on the ETSI MEC standardization, leveraging for instance MEC-native services such as the RNIS. Also the authors in [279] describe a real-world scenario for the development and testing of ADASs with cellular networks. On the same line, the authors in [280] propose EdgeDrive a networked edge cloud services framework that can support low-latency applications during mobility taking into account the needs of the driver, nature of the required service and key network features. Some works consider also AI applied to autonomous cars and to human behavior [281], [282] . Vyas *et al.* [283] state that predicting both driver's stress and behavior is a feature of ADASs systems. Therefore, in their work, they analyze historical trip data to calculate the driving stress and its impact on different driving behaviors. They use LSTM to predict the corresponding stress level of the driver and leverage Federated Learning in a Vehicular Edge Computing architecture, enabling RSUs to do all computing of data (i.e., training) on them. For interested readers, the authors in [284] provide a survey on autonomous driving applications deployed on autonomous embedded platforms and edge devices, focusing especially on energy-efficient approaches for connected autonomous driving, ranging from vehicular communication, edge computing, approximation techniques to novel software-hardware frameworks.

Vehicular clouds. Several works propose to move the computing resources within the ITS users. Zhang *et al.* [212] propose a hierarchical cloud-based Vehicular Edge Computing (VEC) offloading architecture, to reach the optimal computation offloading, considering both the minimization of task delays and the maximization of the network provider's revenue. Similarly, the authors in [285] show how the presence of computing resources in cars could help in sustaining the offloading of computing tasks of low-latency applications. In [211], the authors propose a collaborative MEC scenario depending on the so-called heat zones, where the different degree of heat stands for vehicle density inside a certain area, for vehicular task offloading. To achieve the MEC cooperation they formulate it as a utility maximization problem by designing a non-cooperative

game-theoretic strategy. Through simulations, they show the feasibility of their solution comparing it with several policies. In [157], the authors propose to use FMeC for handling computing problems with the computing power of vehicular clouds (see Section 2.1.2 for further explanations). Other works focus on the possibility of using the vehicles themselves to create a (micro) cloud. In particular, this branch of research is often referred to as *vehicular cloud computing*. The definition was first proposed by Gerla in [286] and it has been further developed in [287]. With the vehicular cloud computing paradigm, a user sends a request to a car using V2V or V2I communications and then the request is forwarded to discover a communication path to a vehicle offering the desired computing service. Afterward, data exchange and computation happen. For instance, Copeland *et al.* [214] describe the AVEC paradigm (automotive virtual edge communicator), which leverages computing resources and advanced technologies that could be present inside vehicles and that could be exploited during emergencies. In [215], Dressler *et al.* leverage parked cars as edge network and storage infrastructure, forming, therefore, a *vehicular cloud*, to boost the performance and scalability of vehicular networks. In this scenario, they propose a protocol called *virtual cord protocol*, to sustain the dynamic of this scenario (with cars that can come or leave) and show that their protocol can sustain this scenario. Hagenauer *et al.* [216] introduce the concept of vehicular micro clouds (a cluster of cars acting as virtual edge servers). These clusters aggregate all the data which are then transferred to the data center in the cloud. In their paper, they propose a map-based clustering, which is then evaluated against different aggregation rates and backhaul technologies. In another paper [217], the same authors deal with two major problems given by this infrastructure: the selection of the gateway nodes and consequentially, the handover procedures. In [218], Dressler *et al.* propose a novel approach called *macro-micro-cloud*, to reduce the communication complexity and improve the QoS, exploiting an additional layer, called *virtual edge computing layer*, between the data center placed at the core of the network and the users which should use the MEC features. This part is called macro cloud, while the micro clouds are clusters of cars.

Infotainment. Finally, some examples of vehicular infotainment are provided. Ndikumana *et al.* [222] propose to serve self-driving cars by deploying MEC resources at macro base stations, Wi-Fi access points, and roadside units for caching infotainment contents near the customers. The same authors, in [223], propose infotainment caching in self-driving cars, where caching decisions are based on passengers' features obtained using deep learning, showing that their approach reduces the backhaul traffic by 61%.

IoT and Smart City

IoT: IoT takes under its umbrella all the devices that can connect to the Internet. Some examples are UAVs, devices for home automation such as lighting, fridges but also Alexa, Google Home, medical devices, and manufacturing devices. They all need different requirements in latency, storage, bandwidth, and security [288], and to support the new possibilities given by 5G such as network slicing [289]. Therefore, the advent of the MEC paradigm seems perfect to help IoT meet all its requirements, as discussed in [290]. However, in this chapter, we do not focus on the MEC support for general IoT, since in literature several surveys already cover this topic (see, e.g., [291], [292] and [293]). We rather focus on new MEC-enabled verticals, one of which is smart city.

Smart City: Indeed, thanks to the increasing importance of the IoT paradigm, also cities are now evolving, installing sensors and IoT devices and therefore becoming *smart* [294]. The collection of data from users, IoT devices, sensors, or more generic devices will allow us to understand deeper which are the critical points of city management and therefore help develop new strategies, to reduce costs, and improve safety and resource consumption. Furthermore, projects such as SmartSantander [193], 5Gcity [232], or SynchroniCity [294] are giving a glimpse of what the city of the future will look like. According to [294], five macro-themes are currently evaluated in most of the smart cities:

- **Mobility:** This includes smart and secure car/bike parking, electric bike usage monitoring, public transportation usage, traffic optimization, and adaptive lighting.
- **Sustainability:** Some examples are noise pollution planning, air quality evaluation, urban waste management, and water management (also called Smart Water).
- **Governance:** for instance, agile governance, environment monitoring, open data accessibility, and citizens' engagements in urbanization.
- **Data Mining:** data lake value extraction.
- **Security:** citizens awareness of IoT.

We now comment on how the literature addresses those macro-themes.

MEC implementation: MEC seems the most promising technology to sustain the smart city paradigm. Indeed, thanks to its *multi-access* paradigm, it will support the connectivity of a variety of devices (GPRS/UMTS/LTE, Wi-Fi, or wired interfaces) altogether. Moreover, it can collect and real-time process, for instance, large amounts of data, and store local information (for security purposes), thanks to its deployed physical edge capabilities. Thanks to the low latency achieved by the MEC presence, a driver could then be informed in a very short time if an accident happened somewhere in the city and which alternatives, he/she could take. Similarly, cameras can perform a first processing of the recorded images at the edge, sending the frames to a central cloud only for special purposes.

Even though the literature on MEC in smart cities is still scarce (most of the papers are magazines), it is possible to draw some directions on the ongoing research efforts. Several papers tackle the issues of MEC implementation in smart cities or even in smart homes, the latter leveraging D2D communications [295]. For instance, in [193], the authors propose a MEC architecture for large-scale IoT deployments (as Smart Cities) supporting existing and future IoT platforms and compliant to the ETSI MEC standard. The authors of [235] propose a smart city scenario in which real-time and time-sensitive applications offload their tasks to MEC servers deployed in cars. They propose an optimization problem to minimize the completion time with a given cost of task scheduling, developing four evolving task scheduling algorithms. Through simulations, they compare them against each other, highlighting that only one (the distributed and improved Jacobi ADMM algorithm) reaches performance close to the optimal.

Recent papers considered also the presence of UAVs in a smart city scenario. For instance, in [240] the authors leverage UAVs for analyzing the position of vehicles in the industrial areas

of the smart city while Xu *et al.* [241] consider an energy-aware multi-UAV task computation management problem for tasks offloading and scheduling according to a realistic Autonomous Delivery NETwork (ADNET). The authors in [242] propose to use MEC to assist electric vehicles in deciding on where to charge or switch their batteries when they are close to finishing their reserve. In this case, the presence of MEC could help by decreasing the amount of information sent between vehicles and infrastructure (i.e., communication cost) and increasing the service satisfaction rate. Thanks to the presence of MEC many application services could be run at the edge of the cellular network. However, services could have different requirements (e.g., low latency, reliability, etc.) and therefore it is fundamental to understand how to efficiently deploy these services to fully support all types of applications. Within this research problem, the authors in [243] propose MAACO, a Mobility-Aware priority-driven service placement model that prioritizes applications according to their criticality and minimizes critical applications' latency, while considering predicted paths for mobile users. In [244], the authors overview the merging of MEC and ICN architecture, giving a possible use case for a smart city scenario. Finally, for interested readers, the authors in [245] provide a survey on general edge computing in smart cities, devising a taxonomy according to different parameters, such as edge analytics, edge intelligence, resources, caching, resource management, characteristics, sustainability, and security.

Machine learning: Furthermore, in smart cities it is important that decisions at MEC level are fast and mostly correct. Machine learning, especially in the form of DRL, seems a promising solution to achieve these goals. In [231] the authors propose a framework that leverages SDN, ICN, and MEC computing capabilities to provide caching and dynamic orchestration of computing resources at the edge. Their goal is to improve the performance of applications in Smart Cities. They developed a big data DRL algorithm and through simulations, they showed the higher performance of their solution in terms of total utility (up to 60%) compared against several schemes (e.g., same scheme but without edge caching or virtualization etc.). Liu *et al.* [233] state that green energy management systems are becoming more and more important due to the development of smart cities. Hence, they develop a model for an IoT-based energy management system, leveraging DRL, on top of an edge computing infrastructure. They compare their solution in terms of delay and energy cost against baseline energy scheduling methods (e.g., only-cloud methods), showing that their DRL-bases method achieves less energy cost (up to 60%) and a smaller overall delay (25%).

Recent papers also use advanced or novel ML techniques. For instance, the authors in [246] use distributed deep learning to perform tasks offloading from IoT devices, considering the presence of heterogeneous computing environments (i.e., edge and cloud nodes). Ale *et al.* [247] state that IoT services demand exhibit spatial-temporal features and therefore it is of uttermost importance to deploy and allocate MEC servers at optimal locations to meet service requirements in a smart city. They propose a spatiotemporal Bayesian hierarchical learning approach to learn and predict the distribution of MEC resource demand over space and time to improve MEC deployment and resource management. Instead, Siya *et al.* [248] propose a RL-based joint communication-computational resource allocation algorithm, showing that it achieves saving energy consumption, reducing processing time, and guaranteeing QoS for 5G applications in smart cities. The authors in [249] focusing on traffic congestion and road congestion, propose a Deep Learning algorithm to build a short-term traffic flow prediction model of 5G IoV while Want *et al.* [250] propose a data augmentation based cellular traffic prediction model using generative adversarial networks to

improve the cellular traffic prediction performance while protecting data privacy and alleviate the negative impact of data insufficiency.

Zhao *et al.* [234] study the always-changing service demand due to crowds in a Smart City. To balance the network load and avoid network congestion and annoying delays, they developed a smart algorithm based on DRL, showing that it achieves better performance than algorithms such as OSPF and EOSPF (from 10% and up to 50%, on average).

Video streaming: smart cities themselves will also be a container where other verticals (e.g., automotive, media, manufacturing) will be merged and further studied. However, in this context, only media has been evaluated in smart cities with MEC so far (especially for video streaming). In [232], the authors explain the 5Gcity project, which has the goal of creating a MEC neutral host platform for smart cities, focusing especially on ultra-high definition video streaming, live streaming, and AR/VR use cases. To show the feasibility of their architecture, they evaluate three different use cases by deploying testbeds in three European cities (Bristol, UK; Lucca, Italy; and Barcelona, Spain), In another paper that tackles video streaming in smart cities, Taleb *et al.* [296] propose the merge of FMeC concepts (evaluated in Section 2.1.2) with MEC capabilities to maintain constant the QoE of video streaming while users move. Specifically, they enable MEC service migration to follow users. Zhao *et al.* [251] propose a hierarchical emotion recognition system using DNN enabled by MEC, since according to them computing DNN tasks in IoT devices will be too costly while the authors in [252] propose a classical joint video content caching and user association in MEC networks, to decrease overall latency and handover latency in smart cities.

Security: Both security & privacy are topics of uttermost importance for smart cities. Indeed, collecting, managing, and processing sensible data at the edge could lead to attacks from malicious users or to data breaches, with catastrophic scenarios. The next discussed group of papers focuses on several security aspects of MEC in smart cities.

In [237], the authors propose a selective recommendation mechanism based on compiling dynamic black- and white-lists, to identify trustworthy participants that can access smart city devices. With data-driven experiments, based on both personal health and air quality monitoring, they show the effectiveness of their solution in avoiding malicious attacks in various scenarios, comparing it also against other similar algorithms proposed in [297] and [298]. Wang *et al.* [238] focus on the security threat given by low-cost IoT devices and the MEC deployment near the RAN. They state that upper-layer cryptography is not feasible for resource-limited scenarios and propose a comparison between information security mechanisms implemented via physical layer approaches. Rahman *et al.* [236] propose a framework that leverages blockchain, AI, and edge nodes to offer secure smart city services (sharing economy, smart contracts, and cyber-physical interaction. Finally, Gheisari *et al.* [239] propose a privacy-preserving architecture, leveraging ontology at the edge network, for IoT devices in a Smart City scenario. Through simulations, they show that the ontology would allow for preserving privacy in a heterogeneous IoT scenario.

Recent studies consider the task offloading problem coupled with privacy/security concerns, leveraging techniques such as game theory [253], [254] or deep learning [255]. In the latter in particular, the authors propose a novel MEC architecture in a smart city scenario to mitigate IoT attacks, using federated deep learning. Finally, some papers use blockchain as a way to increase security and privacy. Lin *et al.* [256] propose a Peer-to-Peer (P2P) computing resource trading system to balance computing resource spatiotemporal dynamic demands in an IoV-assisted smart city,

leveraging blockchain to guarantee privacy and security. They formulate a two-stage Stackelberg game, evaluating their proposal through numerical experiments. Similarly, Ye *et al.* [257] study a task offloading IoV scenario with MEC nodes deployed in a smart city context using blockchain for added security. They formulate an optimization problem and a MDP, proposing then a deep reinforcement learning algorithm.

Table 2.4: List of papers related to media, manufacturing and eHealthcare verticals

Industrial verticals	References	Use Case	Tools	Evaluation	Most relevant lessons learned
Media	[299], [300], [301] [302], [303], [304] [305], [306], [307] [308], [309], [310] [311], [312], [313] [314], [315], [316] [317], [13], [318] [319], [320], [321] [322], [323], [324] [325], [326], [327] [328], [329], [330]	- ABR video streaming with MEC - Cache placement - Block chain video streaming assisted by MEC - QoE enchantments - Cooperative video processing - AR/VR/XR support - Metaverse - UAV assisting video streaming - Digital Twin	- (M)ILP - Optimization - Auction frameworks - Dynamic programming - Optimal matching theorem - Multipath routing algorithm - Lyapunov theory - Machine Learning - DRL - Lagrangian optimization	- Numerical simulations - Network simulations - Testbed performance evaluation	- Caching with MEC will improve backhaul traffic load and average access delay with respect to established approaches - MEC, together with fiber-wireless access networks, will outperform the Mobile Cloud Computing paradigm in terms of RTT latency (up to 50% of difference) - MEC could support VR in in terms of latency reduction (compared in scenarios w/o MEC) and energy efficiency - MEC processing of VR tasks will decrease the traffic in core and radio access up to 80.5%)
Manufacturing	[331], [332], [333] [334], [335], [336] [337], [338], [339] [340], [341], [342] [343], [344], [345]	- Resource scheduling for manufacturing - Multi-tier MEC for satisfying IIoT requirements - Task offloading - Avoiding deadlock in resource provisioning - Digital Twin - Automated Guided Vehicles (AGVs) - Support smart factory process	- DQN - Multi-agent DRL - Two-step algorithm - Deadlock avoidance algorithm - D2D offloading with MEC	- Prototype evaluation - Numerical simulations - ETSI-MEC compliant simulations	- MEC's proximity will decrease computing delays (up to 40%) and energy consumption. - Some examples of devices affected by the energy minimization are AGVs and/or robots in smart factories
eHealthcare	[346], [347], [348] [349], [350], [351] [352], [353], [354] [355], [356], [357] [358], [359], [360]	- Abnormal patter detection in patient's state - Comprehensive MEC architectures for smart health - EEG- based pathology detection system - Blockchain based health monitoring - Monitoring COVID-19 - Internet of Medical Things	- Feature extraction - Signal processing - Tree-based deep model - Bloom filter - Deep Learning - Stackelberg game optimization - Blockchain - Sample Average Approximation - D2D	- Numerical simulations - Testbed experiments	- MEC will help health applications in a wide range of fields, from Data reduction, bandwidth and energy saving and low latency - MEC will reduce the latency up to 50% compared to cloud-based networked healthcare systems - Different computing models could support distinct eHealthcare use cases

Media

As of today, 70% of the overall data traffic is owned by video applications, e.g., it comes from platforms like *Netflix*, *YouTube*, and *Twitch*. In the next years, this share is expected to grow due to the

advent of virtual and augmented reality applications. These applications impose tighter constraints than other video applications, especially in terms of delays, bandwidth, and computation [13], [361]. Therefore, both for canonical video streaming and AR/VR, it is of vital importance to move resources at the edge of the network, leveraging the new MEC paradigm.

Video streaming: In this context, MEC will be useful for increasing the overall Quality of Experience (QoE), exploiting several approaches such as caching, cooperation between MEC nodes, and offloading of heavy computational tasks (e.g., transcoding), even merging these concepts. For instance, Tran *et al.* [299] propose to leverage collaborative MEC servers to enhance video caching and processing support for adaptive bit rate (ABR) video streaming. This collaborative joint caching and processing problem is formulated through an integer linear problem, to minimize the average access delay to video users. To address this problem, they formulate a low-complexity online request. They use simulations to show that their approach outperforms by $\approx 20\%$ caching techniques such as Most Popular Caching and other schemes [362]. The authors of [302] study the caching at the edge for improving the QoE of live video streaming. They propose two auction frameworks for the caching space allocation at backhaul (Edge Combinatorial Clock Auction and Combinatorial Clock Auction in Stream), showing via simulations that they achieve higher performance, about 10% better if compared to baselines.

In [300], the authors design a scenario for video streaming with MEC resources, studying how fairness (of edge computation capabilities) and QoE can be improved with MEC against baseline client-based DASH heuristics. Using a network simulator (SimuLTE) they show the superiority of their scenario in terms of bitrate per client (20% higher on average), initial buffer delay ($\approx$ 15%-20% smaller), and Jain's fairness index [363]. The goal of Long *et al.* [364] is to improve the detection accuracy of human presence using cameras. They leverage cooperative MEC nodes for pre-processing tasks. Their focus is especially on how to partition video tasks and how to match tasks to edge nodes. The MEC, thanks to its edge computing resources, can exploit tools such as machine learning and blockchain to support QoE improvements. The authors of [304] propose a proof of concept based on LTE for MEC support for mobile video streaming. The MEC server caches popular videos and, based on the radio condition, chooses the most suitable video quality. They further propose two machine learning algorithms for popular video prediction and forecast of channel quality. Through numerical simulations, they show, for instance, that the prediction model for radio channel quality reaches over 80% of prediction accuracy. Instead, Liu *et al.* [301] propose a blockchain video streaming framework assisted by MEC, where heavy computational tasks such as video transcoding can be offloaded to MEC nodes. They compare their solution against the same one without the blockchain component, showing that the latter performs worse, up to 35%, in terms of average delay.

Finally, several papers tackle MEC implementation with real LTE testbeds, to support video streaming. Martin *et al.* [303] design a new MEC component for video streaming called MEC4FRE. This application retrieves data analytics from layers 2 (RAN awareness), 3 (media delivery metrics), and 7 (MPEG-DASH manifest for local caching) to dynamically prevent QoE degradation and keep radio efficiency high. The authors compare their solution against a best-effort delivery strategy in a real LTE infrastructure, where they proved that their solution achieves better performance.

Ge *et al.* [305] present a novel MEC real-time QoE estimation VNF, which has been imple-

mented and deployed in a real LTE-A network edge. They show that their VNF can correctly estimate QoE in real time and its CPU and RAM usage are both very low.

Recent papers continue to focus on increasing the overall QoE for end-users, using a variety of different techniques. For instance, the authors in [311] use a reverse-fuzzy particle swarm optimization algorithm for QoE-aware offloading problem, while Shi *et al.* [312] consider a QoE-aware MEC selection scheme to select the best MEC to serve end-users. The authors [313] formulate a mixed integer nonlinear programming (MINLP) problem to minimize the video service latency in a joint caching, computing, and power allocation problem. To solve this problem, they transform the problem into an Integer Linear Problem and then which is thereafter solved by MATLAB intlinprog function. Chen *et al.* [314] jointly address the caching placement, video quality decision, and user association problem in the live video streaming service coupled with MEC and formulate a NP-Hard problem which is then solved with a Lagrangian optimization. Also in [315] the authors consider a joint optimization problem of video segment caching and transcoding in MEC servers and resource allocation to improve the QoE. Again, their problem is NP-Hard and therefore they propose a low-complexity heuristic. Some recent papers considered also the presence of UAVs assisting MEC to support video streaming or other video-related actions (e.g., transcoding) [316], [317]. Interested readers, who want to deepen how MEC could support video streaming should also check the following updated surveys [365][366].

AR/VR: Thanks to the recent technological hardware advancement, more and more realistic Virtual Reality (VR) and Augmented Reality (AR) applications are present, notwithstanding the demanding bandwidth and delay requirements. According to Huawei Technologies and the China Academy of Information and Communications Technology (CAICT) [367], to achieve the entry level of immersion experience in VR, with a 4K 2D video, the bandwidth provided to the service should range between 20-50 Mbps with a round trip time (RTT) latency of maximum 40 ms. Instead, for a full immersion experience (with a 24K 3D screen), the bandwidth should range from 2 to 5 Gbps and RTT below 10 ms.

Indeed, a recent paper showed how supporting VR would be impossible for current 5G networks and even for beyond 5G networks [13]. However, it is important to study novel techniques that could be applied in future scenarios with next-gen technologies. For instance, Du *et al.* [318] consider using bandwidth-rich terahertz (THz) communications to support the offloading to MECs of the viewport rendering for high-quality immersive VR video service while the authors in [319] study how cache placement in several MECs could help to sustain the end-to-end latency for VR video delivery.

Indeed, according to [368], MEC features such as high proximity computing, proactive caching, and support to mmWave are needed for AR/VR successful delivery, taking into consideration also that computing and communications delays are the two most relevant bottlenecks in AR/VR cases. Hence, a MEC deployment becomes of primary importance. The authors of [306] propose an integrated heterogeneous networking scheme, taking into consideration the fiber-wireless access networks, using a virtualization technique to achieve the demands of the applications. They evaluate their solution with a testbed, showing that this infrastructure supports the AR/VR requirements and outperforms other paradigms such as Mobile Cloud Computing in terms of RTT latency (with differences up to 50%). In [308], the authors define the main challenges for a full wireless interconnected VR (Quality-rate-latency tradeoff, Localization and tracking accuracy, green VR among

others). Further, they focus on three possible interconnected VR study cases: the first is about leveraging the joint resource allocation and computing, the second one shows the benefits stemming from exploiting proactive computing against reactive computing, and the last one studies an AR-enabled case with self-driving vehicles. With simulations, they show that with nowadays technologies it is still impossible to reach a fully interconnected VR scenario. Similarly, the authors in [309] argue that most of the works in this area consider only computation-constrained MEC scenarios, neglecting the communication perspective. Therefore, they propose a MEC framework to reduce communication resource consumption by leveraging caching and computation resources of VR devices. They formulate an optimal task scheduling policy to minimize the average transmission data per task. Through numerical simulations, they show that it achieves higher performance in terms of average communication costs ($\approx 45\%$) compared to baselines. Immersive videos for VR, also known as **360-degree** videos, provide an interesting VR feature, thanks to the omnidirectional view they offer. Several papers tackle the use of MEC for immersive videos. Liu *et al.* [369] develop a multi-connectivity scenario for 360-degree videos (MEC's computing resources for active transcoding and caching + mmWave/sub 6 GHz for supporting high bandwidth VR). Furthermore, within their scenario, they formulate a novel communication and computation resource allocation problem. Through simulations, they compare their solution against cases in which some technologies were not present, showing that it achieves better performance in terms of latency and energy efficiency (from 15% up to 25% on average). In their paper, Sun *et al.* [310] model several trade-offs between communications, caching, and computing with MEC in a mobile 360-degree VR scenario. They first propose a novel MEC framework for this scenario and then formulate an optimal joint caching and computing policy to minimize the average transmission rate, under several constraints (latency, cache size, and average power consumption constraints). They obtain a closed-form expression and evaluate it against several greedy algorithms, showing that it achieves higher performance (depending on the scenario, from 30% to 50%). Mangiante *et al.* [307] propose a rendering solution for 360-degree videos leveraging MEC, to optimize the latency and bandwidth resources. Through preliminary tests, they show the benefits of having an edge network infrastructure in terms of reducing by up to 80.5% data traffic delivered towards a centralized cloud and radio access.

Recent papers considered also using machine learning techniques such as RL [320] and DRL video [321].

XR/Metaverse: Two novel concepts recently introduced are the eXtended Reality (XR) and the Metaverse. The first one is an umbrella term that describes immersive technologies that can merge the physical and virtual worlds (i.e., reaching for instance the digital twin paradigm). Under its term, it blends AR, VR, and Mixed Reality (MR). Within this paradigm, at the moment researchers focused on applying ML techniques such as DRL to support the offloading of XR tasks to MEC/cloud systems [322], [323], or using a cooperative non-orthogonal multiple access (Co-NOMA) scheme to support several XR devices [324]. Instead, regarding the Metaverse, which refers to creating virtual worlds where users control and interact with avatars, researchers focused on different aspects such as proposing novel architectures to support the stringent requirements of the metaverse [325], or creating novel QoE metrics, such as the "meta-distance" (i.e., to measure both the service delay and social distance among metaverse users) [326]. Others focused on the digital-twin aspects of the Metaverse [327], [370] on localization (e.g., how MEC could help in decreasing the end-users localization errors [328]) and finally Yao *et al.* [371] leverage blockchain

to create an assisted secure cross-metaverse authentication scheme in a MEC-metaverse enabled scenario. For interested readers about the Metaverse, two interesting surveys have been recently published [329], [330].

Finally, we mention that an updated review of some aspects of this vertical has also been presented in Section 1. In that section, we reviewed recent papers on cloud/edge gaming and MAR, with a particular emphasis on jobs/tasks offloading and energy aspects.

Manufacturing

In 2019, the 5G Alliance for Connected Industries and Automation (5G ACIA)[6] was created. Its goal is to apply 3GPP 5G specifications for Smart Factories [372] to the operation of manufacturing and processing industries. 5G ACIA has six working groups, covering aspects like architecture and technology for industries, use cases and requirements, and spectrum and operating models among others.

Smart factories are context-aware systems that "assist people and machines in the execution of their tasks" [373]. The context includes the status and position of an object based on virtual and physical information available, enabled by both machine-type communications and IoT devices.

The MEC is an important means of implementing some of the key design principles introduced with the Industry 4.0 paradigm. In particular, it paves the way towards interoperability of machines, virtualization of physical resources, decentralization, and real-time capabilities in the analysis of data (thanks to the support of VNF, 3rd party, and industrial applications). The use of MEC also helps in terms of achieving low delays, which is vital for some IIoTs applications that tolerate no more than 250 μs delay [374] (such as robot motion control and packaging machines).

In real case examples, the MEC might have access to all the processes in a Smart Factory, from logistics to supply and inventory management. The MEC might therefore be able to retrieve data from all the sensors of IIoT devices, and automatically and dynamically make decisions according to a predetermined goal.

MEC infrastructure: Due to the diversity and complexity of factories in terms of production, machinery, spaces, and specialized workforce, the MEC infrastructure needs to be carefully designed to allow the proper level of flexibility for smart manufacturing plants. A first attempt to provide a specific MEC infrastructure for smart factories is in [331]. The authors propose a 3-level hierarchical smart factory architecture, in which they highlight a physical resources layer, a network layer, and a data application layer. The first layer contains all the manufacturing resources that, through sensors and RFID (among others), can interact with the second level. The latter includes networking technologies such as access points access and switches (deployed according to new paradigms like MEC and SDN). Finally, the third layer allows for the analysis of the retrieved data to gather useful information about the status of the smart factory, to be sent to end users (workers or engineers). Similarly, Dao *et al.* [333] propose an mMEC, i.e., a multi-tier MEC architecture keeping in mind several IIoT challenges such as the processing of big IIoT Data with ultra-low latency and reliable response, and context awareness. Finally, the authors of [332] propose a hybrid computing solution framework, to propose a resource scheduling strategy for

[6]https://www.5g-acia.org/

real-time smart manufacturing applications in an edge computing scenario. Through a prototype implementation, they show that their strategy outperforms other approaches, e.g., centralized cloud, in terms of computing latency (≈ 15%-33% on average).

Reliability: This is a topic of uttermost relevance in smart factories. IIoT devices need > 99,999% of successfully transmitted packets, to avoid malfunctioning in the production lines and accidents that could harm workers. The following papers provide the most relevant examples of issues that impact the overall reliability of a MEC system in smart factories. The authors in [335] study a resource request banker's algorithm to avoid deadlocks that could occur in the presence of several IIoT devices accessing the MEC resources (a behavior that they confirmed through simulations). With their algorithm, they prove that the probability of a deadlock in a MEC scenario will be reduced up to 12% compared to a scenario without any deadlock avoidance algorithm. Luo *et al.* [336] propose an adaptive task offloading auction mechanism that allows Industrial Cyber-Physical Systems (ICPS) to offload their tasks to several MEC servers chosen based on task deadlines and the required security levels. Using simulations, they show the superiority of their approach compared to baseline schemes using randomized and FIFO scheduling. Finally, the authors in [334] propose a two-tier partial offload MEC-cloud framework in a heterogeneous energy-constrained IIoT scenario to optimize the transmission reliability and IIoT energy consumption. They formulate a low-complexity solution and evaluate it through simulations. They compare their algorithm against two baseline solutions, showing that it achieves higher performance in energy consumption and blocked devices (from 10% to 20%).

Recent papers tackle several aspects of smart factories. For instance, Hsu *et al.* [337] propose a two-tier MEC architecture for the partial task offloading in a computation, and communication (of licensed and unlicensed bands) resource allocation problem, considering both energy efficiency and QoS satisfaction. In [338], the authors also study the problem of task offloading coupled resource scheduling. In this case, the scenario is multiple Automated Guided Vehicles (AGVs) performing smart factory patrol service and the authors' goal is to minimize the overall energy consumption of the AGVs, by jointly using MEC and D2D offloading. Other papers propose using Machine Learning. For instance, in [339], the authors use DQN to schedule tasks from a smart factory process in an edge-cloud architecture, while Cao *et al.* [340] propose a Multi-agent DRL approach for the multichannel access and task offloading problem in a MEC-enabled industry 4.0 scenario. Some papers consider the ETSI MEC standardization paradigm in their work. For instance, the authors in [341] implement in ns-3 ETSI MEC entities and functionalities to allow the simulation of smart factory scenarios with an ETSI MEC-compliant architecture. Instead, Borsatti *et al.* [342] propose an ETSI-MEC compliant architecture to support the automated deployment of Industrial IoT applications at the edge while in [343], the authors want to satisfy the requirements of industrial applications using traffic steering, using ETSI-MEC nodes for packet payload inspection and processing. The authors in [344] propose a MEC framework to support the instance of an IoT service layer at the network edge to enhance the QoS of IoT applications, showing as an example the remote control in manufacturing. Lee *et al.* [345] instead propose an integration of the blockchain concept in MEC systems to support smart manufacturing systems. Finally, for interested readers, the authors in [375] surveyed how MEC-empowered network slicing solutions could sustain IIoT scenarios.

eHealthcare

Another important vertical that is gaining attention is eHealthcare. Medical tools are becoming more and more sophisticated, with multiple sensors and data (ranging from video, signals, and personal data) that have to be processed. Moreover, consumers are paying progressively more attention to well-being, with an increasing demand for quality devices, safety, and data storage. Therefore, these requirements bring the necessity to move computational resources closer to devices, to perform faster, efficient [376] and accurate decisions.

Edge nodes can also be leveraged for performing data pre-processing, to send only selected data toward a centralized cloud, helping in both reducing bandwidth utilization and improving privacy. On this line, the authors of [346] study an abnormal pattern detection mechanism of a patient's state at the edge of the network, where edge nodes send only the most important features in a centralized cloud. Further, in case of detected anomalies in the patient's state pattern, it pings the nearest healthcare provider. In another paper, the same authors enhanced the framework proposed earlier with the MEC architecture [350], highlighting the benefits that MEC will bring in several smart health applications (for instance low latency for real-time epileptic seizure detection or prediction of bradycardia in preterm infants or reducing bandwidth allocation for continuous services such as remote cardiac monitoring or Parkinson's disease detection). Similarly, in [347], the authors leverage the MEC for a preliminary data processing of electroencephalogram signals (for smart pathology detection) before sending the data to a centralized cloud. Pace *et al.* [351] propose to create an edge layer between cloud and IoT devices belonging to end users, to reduce communications delay and increase privacy levels. They evaluate their framework with a real test bed in two different scenarios (workers in a factory and athletes in a fitness center), showing that their framework would reduce the communications delay and the overall data transmitted to the centralized cloud by 20%-50%. In [349], the authors propose to collect health information to monitor patients' health via UAVs and then process the data in MEC servers (possibly in the nearest one) leveraging blockchain to increase data security. Through simulations, they show the effectiveness of their scenario. Chen *et al.* [348] describe a cognitive edge computing smart-healthcare system, with the double goal of evaluating the patient's health using an edge cognitive computing paradigm and, depending on the health-risk grade of each patient, allocating edge communications resources to better assist them in emergencies. Furthermore, in [352], Muhammed *et al.* propose a framework called UbeHealth, which leverages edge computing, deep learning, big data, and high-performance computing to support healthcare systems in smart cities. They developed a proof of concept and performed an evaluation based on a nationwide networked healthcare system with three different data sets. They show that, with their proof of concept, latency is reduced by 50% compared to cloud-based healthcare solutions. Finally, Li *et al.* [377] present Edgecare, a secure and efficient data management system, to improve the management of decentralized healthcare data, leveraging edge computing paradigms such as MEC. They propose an optimization problem and, through numerical simulations with security analysis, showed the effectiveness of their framework.

Novel papers also tackle the eHealthcare scenario. The authors in [353] propose a task offloading problem in an IoT-eHealth scenario where devices could also migrate (and therefore tasks allocated in edge servers should be migrated). They propose a Multi-Stage Stochastic Programming formulation and solve it using a Sample Average Approximation algorithm. COVID has

widely impacted the lives of millions of people and researchers started to study this phenomenon correlated with MEC. Feriani *et al.* [354] propose a hierarchical MEC framework to monitor the physical conditions of human subjects and COVID-19 symptoms in particular (e.g., fever, coughing, and fatigue) while in [355] the authors envision several use cases for realizing contact-less approaches that assist the mediation of COVID-19. Suraci *et al.* [356] proposes an eHealth system architecture, in which low-latency enabling technologies like Device-to-Device (D2D) communications and MEC are integrated and supported by security mechanisms. Their goal is the optimal management of sensitive health data collected by Internet of Medical Things (IoMT) devices. Zhang *et al.* [357] develop an algorithm that employs a forecasting model to extract user behavior characteristics and quantifies the forecast results reasonably by introducing queuing theory, providing a basis for the matching of resources and users. Blockchain is also studied in this domain, especially for security aspects. In [358], the authors design a data-sharing scheme, which enables data exchanges among healthcare users by leveraging blockchain and interplanetary file systems. Particularly, they integrate a smart contract-based authentication mechanism with MEC to perform decentralized user access verification without requiring any central authority. In [359], the authors propose a Combinatorial Auction and Improved Particle Swarm Optimization Computation Offloading Approach (CA-PSO) for the offloading at the network edge of low-delay healthcare information given by Internet of Medical Things (IoMT) devices. Alekseeva *et al.* [360] study how three computing models (local computing, MEC, and Mobile Cloud Computing) could support several eHealthcare use cases such as Remote medical examination, robotic surgery, and cardiac telemetry. For interested readers, the authors in [378] delineate future research efforts for the eHealthcare vertical in 6G networks.

Summary, lessons learned and open challenges

During our literature review in 2019 we discovered that, in general, so far MEC had not been fully evaluated for vertical industries. Most of the papers reviewed were architectural, with few of them that analyzed real datasets or evaluate performance figures of real devices.

Looking at a more general perspective, Tables 2.3 and 2.4 show that the most studied verticals were automotive, smart city, and media. With the help of the table, we next updated the review on lessons learned and open research challenges for each of the verticals.

Automotive: Focusing firstly on the automotive domain, we see that MEC is considered a fundamental building block for achieving efficient C-V2X communications and novel use cases, thanks to its possibility to achieve low latency [259], [260]. The main takeouts can be summarized as follows:

- 5GAA has identified four possible use case groups: safety, convenience, advanced driving assistance, and VRU. Important research efforts have been devoted to safety and VRU ([206]–[210], [224]–[226], [228]–[230]), showing that the MEC presence, thanks to its proximity to the end users and high computation power and pervasivity [228], will be of great help to improve both vehicle and pedestrian safety (for instance by offloading the computation of collision detection algorithms to close MEC servers). Indeed, according to to [209] 100% of collisions with autonomous cars could be detected on time, while with human drivers the number slightly decreased by 14%. Notwithstanding, particular caution

should be made to users who are changing the edge server to which they are anchored [224]. Finally, some works also provided results and considerations from real-world experiments, showing the feasibility of MEC for VRUs in real case scenarios [225], [226].

- However, offloading decisions are not trivial to make, since they should also consider the presence of a possibly high density of vehicles [211], and revenues generated by different vehicles [212]. Novel metrics and tools could help study the network performance, such as AoI [229]. Due to the scarce presence of edge nodes, also the orchestration or resources is very important [230], [379].

- MEC will also help to provide infotainment to drivers and passengers ([125], [222], [223]), especially leveraging caching together with deep learning, which allows reducing backhaul traffic by 61%.

- Most of the available papers have identified several technical challenges, such as enabling edge communications ([213] proposes to use three different access technologies), or ad hoc computing resources such as vehicular clouds [214]–[216], [218]. The vehicular cloud paradigm allows computing resources even within vehicles, pushing the MEC paradigm to the very edge. This scenario however imposes tough challenges due to its volatility (for instance, a car might join the cloud at any time). Specifically, data management and communications between clouds and backhaul (both in uplink/downlink) become cumbersome, needing, therefore, more in-depth research effort.

- Furthermore, it emerges that the MEC also supports pioneering assisted-driving applications, such as platooning. Several papers addressed this topic, showing that MEC offers a possible solution to sustain this paradigm. For instance, [219] and [220] propose an architecture for managing platoons and avoiding shockwaves, [265] focused on offloading decisions, [221] proposed a MEC that can form and coordinate platoons. [266], [267] showed that a MEC centralized control of speed and acceleration of platoon vehicles is a viable alternative to V2V communications. Some recent papers also studied the support of platoons through a standard ETSI MEC architecture [268], [269] while others considered the general problem of task offloading in a platooning scenario (leveraging edge servers or the presence of computation in vehicles in the same platoon [273]. Some works even studied using UAVs to support the formation and control of platoons [271], [272]. With the advancement of new technologies, also new papers consider different assisted-driving applications. [270] and [275] for instance focused on developing tools for creating high-definition maps to support vehicles or platoons, while in [276], the authors presented an architecture to support a Cooperative Autonomous Driving Maneuver Control application for the cooperative lane change. Other works [277]–[280] presented solutions to support ADASs while a handful of papers use Machine Learning techniques to support assisted-driving use cases [281]–[283].

Notwithstanding the large amount of published papers, still many open research challenges remain, e.g.:

- **Security** is the uttermost theme to be developed in vehicular networks assisted by MEC. Indeed, with the growing possibility of having more connected cars and edge resources

on the road, there are also more possibilities for malicious attacks. These scenarios must be avoided and therefore research should focus more on the security aspects of this new paradigm that embraces connectivity and computation. Some examples of risky procedures in the MEC environment are migration of resources (VMs and Containers), MEC deployment billing, and what refers to the coordination of multiple new nodes introduced with the MEC architecture [199]. Some recent papers considered those security aspects [380], even for instance relying on blockchain [381] but more works are needed.

- More work should also be oriented to **VRU** and general **safety**, with the development of new collision detection or avoidance algorithms, also leveraging on prediction techniques given by machine learning which take into account both physical resources and wireless channels [382]. In particular, novel machine learning techniques such as federated/distributed learning could help the development of lightweight algorithms, since the training and validation of these approaches could be done in local servers, thus enabling fast training and also a secure one, since data will be confined in the local vehicles/devices/servers [383]. Also the smart deployment and management of resources (e.g., VNFs, containers, MEC services etc.) [379] could play a role in supporting safety use cases, with also the goal of ensuring reliable connectivity.

- According to Intel,[7] a single autonomous car could generate up to four terabytes of data each day. Hence, the MEC should be able to handle and process that amount of data. Which is more, the MEC should support multiple autonomous cars at the same time. Therefore, **big data** processing and analytics is of fundamental importance for both connected cars and MEC paradigms. However, only a few works [275], [278] have so far addressed jointly these issues in a vehicular scenario. A possible idea could be to make use of cooperation between federated MEC nodes [384] to achieve better results. Another solution could be to leverage **Edge AI** for real-time processing of sensor data, such as cameras and LiDAR, to improve safety features like collision detection, lane-keeping, and adaptive cruise control.

- With the possibility to deploy computing resources at the edge, new **business** opportunities arise, together with the possibility to increase **revenues**, in multi-operator scenarios [262]. While [212] provided a first example of a possible MEC-based revenue generating system and in [385] the authors showed how deploying different MEC clusters could lead to different monetary cost, more research is needed to cover the complexity of this scenario fully.

- Many examples provided by 5GAA [262] and [386] have not been fully studied and evaluated yet: while some examples have been studied (such as real-time situational awareness and handling high-definition maps [270], [275], others have to been fully evaluated yet in an ETSI MEC supported context (e.g., see-Through, self-driving cars, traffic management and control systems). Moreover, future work should also consider 5G features such as network slicing, or the support for new internet architectures, such as ICN [387]. Also, O-RAN will become very important for future cellular network deployments and therefore it would be interesting to study how O-RAN can support the automotive verticals together with edge computing. Finally, it would be interesting to see how MEC could support human-machine

[7] https://newsroom.intel.com/editorials/krzanich-the-future-of-automated-driving/

interaction (e.g., allowing edge-based natural language processing, gesture recognition, and augmented reality interfaces for a more intuitive and responsive in-car experience.

- Finally, the car manufacturing world is slowly shifting from traditional oil-based vehicles to **electrical/hydrogen** vehicles. It would be interesting to study how and if this shift would also affect the MEC support for vehicular networks, and if the MEC could play a role in making cars *greener* and more efficient, and smarter in general. In particular, energy efficiency will play a big role in future networks, and creating algorithms that not only achieve good performance but are also energy-efficient (e.g., meaning for instance that they require low computation at the vehicle's side) would be important. Finally, for interested readers, many novel surveys have been published tackling several aspects of automotive, with some of them considering also the presence of edge computing [263], [388], [389].

Smart city: While IoT as a macro concept has been widely studied, what it has not been fully explored yet is the MEC implementation in smart cities, where the MEC can play a fundamental role in the communication part. Indeed, while the smart city paradigm embeds different verticals (e.g., Automotive, Media, eHealthcare) altogether, it poses new challenges and constraints due to its enhanced IoT deployment nature. For example, in the SmartSantander case, more than 20000 sensors (between fixed, mobile, and smartphone ones) and 2500 RFID tags [193] have been deployed in a Spanish city (Santander), posing, therefore, scalability and QoS challenges (for instance, how to avoid that collision between packets coming from hundreds or thousands of devices would degrade the throughput significantly).

Below are listed some lessons learned and open research challenges:

- First of all, most of the papers surveyed are magazines, even though recently some papers started to appear in journals, mostly in the IEEE Internet of Things Journal. While they give a great overview of most of the possible technical scenarios for smart cities, they lack an in-depth technical view, which instead is needed to better study this vertical.

- Several papers identify the need to define new framework architectures to support this vertical. [193] proposed an architecture compliant to the ETSI MEC architecture, to support enchanted IoT deployments, [231] merged MEC, SDN, and ICN for caching at the edge while [232] showed the 5Gcity project, aiming to develop testbeds for UHD video streaming in smart cities. More recently, in [244], the authors provided an updated overview on the merging of MEC and ICN architecture, giving as an example a smart city scenario while the authors in [255] considered a novel MEC architecture to mitigate possible IoT attacks.

- Many papers claim that optimization techniques and machine learning are key to finally deploying smart cities ([231], [233]–[236]). Both [233] and [234] show that using deep reinforcement learning for energy management systems at the edge and network load balancing in the presence of moving crowds can outperform traditional approaches (e.g., only centralized cloud methods) and algorithms (such as OSPF and EOSPF) from 10% up to 60%. [235] shows how optimization problems for offloading tasks are crucial for real-time-sensitive applications and [236] unveiled the advantages of blending blockchain, AI, and edge computing to support sharing economy services. ML techniques could also be used

to predict short-term traffic for traffic and road congestion [249], [250], to distribute tasks offloading and/or resource allocation [246], [248]

- Finally, several papers point out that severe security issues are unresolved. [237] shows that selecting trustworthy participants for accessing smart city services is desirable, while [238] points out that the main information security challenges are in the physical layer. [239] warns against the lack of suitable privacy-preserving mechanisms. However, several novel research approaches considered security and/or privacy issues in their work, even though in a task offloading/scheduling scenario [253], [254], [257].

Indeed, while the concept of smart city has been theorized many years ago, more technical work is still needed to make it real:

- As for the other verticals, increasing connections between users and things in a smart city context gives hackers the possibility to obtain important personal data, both directly (social security numbers, bank accounts, etc.) or indirectly (by inferring political or religious preferences, etc.). Attackers could leverage the weaknesses of the network infrastructure. Hence, more comprehensive work on **security** and **privacy** issues should be performed, maintaining both a full stack overview and aiming at lightweight solutions, which could be deployed on simple objects with the help of the MEC. Furthermore, we believe that research should specifically consider security physical attacks such as power cutout, fire, and link break [199] (due to the presence of a high population density scenario) creating a more resilient distributed system. Several recent works in other verticals leveraged blockchains to assure security and privacy and naturally seems a promising paradigm to leverage also in a smart city scenario.
- **Machine learning** would be a useful tool to predict crowd/vehicle movements or network traffic, e.g., to avoid congestion. Among all the techniques, federated learning seems the most promising one to preserve users' privacy, since it allows the decentralization of data by only exchanging encrypted machine learning parameters between edge nodes and a centralized server. Recent works started to leverage advanced machine learning techniques (e.g., Federated or Distributed Learning), but there are still many open issues such as how and where to gather and keep the data, and where to process it (e.g., where to perform the training and inference part). Similarly, as already discussed in Section 2.1.2, **Green MEC** systems should be considered to reduce costs for operators or even for public administration entities. For instance, edge nodes could control or gather data from smart grids, performing therefore environmental monitoring.
- While the papers surveyed cover a quite wide spectrum, however, many real use cases are still unexplored [294]. Some interesting examples are how to deal with waste management in real-time through smart grids, the use of smart management, or the synchronization of traffic lights given the presence of crowds and vehicles, leveraging also on ML techniques. Other use cases would be to support users' mobility, since only few works addressed it [242], Public Safety and Security, supporting smart healthcare applications within urban environments [251], Human-Centric Edge Applications that enhance the quality of life for city residents MEC, thanks to its computing capabilities and user proximity, will be a possible enabler for these use cases. Researchers should also consider merging several verticals,

like media or automotive, together with smart cities to provide a more realistic scenario. In general of the five macro-themes discussed before (mobility, security, sustainability, governance, data mining) only the first three themes have been touched by researchers.

- Most of the works are theoretical: it would be interesting to leverage real testbeds such as AWS Green Grass or Azure IoT Edge to compare the performance in real-case scenarios. Moreover, cities are now becoming smarter and smarter, and coupled with the first presence of edge nodes in smart metropolitan areas, it would be interesting to see works tackling real-world case scenarios, datasets, and deployments.

- Researchers should also address these fundamental problems: how to provide scalability with a high number of IoT devices (the SmartSantader project deployed more than 20000 sensors), interoperability between several propriety interfaces (for instance how to allow communications and cooperation between AWS Green Grass, Azure IoT Edge, and the ETSI MEC framework), study how to develop new business models (since this scenario gives new revenue opportunities to operators), or how to support network slicing for multiple tasks or verticals on a shared MEC server/infrastructure or support smart caching at the edge.

- As a possible smart city sub-case, MEC together with smart homes has not almost been evaluated yet. Nowadays, our homes are welcoming more and more “smart" devices (e.g., TVs, home automation devices, vocal assistants such as Alexa, and so on). MEC would help those appliances in several ways: from contents cached in close MEC servers to improve QoE to IoT Data pre-processing at the edge (leading to less information sent over the internet, with implications on users’ privacy), to support of integration in the local smart grid.

- Finally, an interesting paradigm in smart cities is UAV communications with MEC. Indeed, UAVs (commonly known as drones) are becoming more and more powerful while at the same time, their costs are decreasing. Nowadays, UAVs are exploited in many different fields ranging from weather monitoring, and precision agriculture, to package delivery and traffic control [390]. Therefore they are also evolving the concept of a smart city into a bigger smart “metropolitan” area (see Section 2.2). MEC together with UAVs enhance computing offloading at the edge (with a UAV-based MEC server that computes users’ tasks) or helps UAVs themselves during heavier computing tasks (particularly helpful since in most cases UAVs batteries have a limited battery life) [293]. On the same line, LEO satellites could fulfill or extend the UAVs’ role in enhancing the computation-communication connectivity in smart metropolitan areas. Finally, MEC can exploit the O-RAN architecture to better support UAV communications (e.g., to allow radio resource allocation for UAV Applications or flight path-based dynamic UAV resource allocation [391]). For interested readers, we mention more focused surveys related on MEC together with UAV communications ([292], [293], [390]).

Media: MEC will also help in the development of new reliable video streaming connections and in the improvement of AR/VR applications, which impose very tight requirements on bandwidth and latency. The key points we discovered are:

- To improve the QoE of video streaming, a few approaches are beneficial: leveraging caching thanks to the new MEC computing capabilities [299], [300], [302], [313]–[315],blockchain ([301]), cooperation between MEC nodes [299], [364], offloading of heavy computational tasks such as adaptively adjusting bitrate or transcoding [299], [301], [303], [364], leveraging technologies such as UAVs to assist video streaming [316], [317] and machine learning techniques to forecast the channel quality [304]. The MEC will help in improving performance from 20% up to 35%.

- Many works provide also insights on performance within real LTE infrastructures [303]–[305], showing that the MEC presence, even just in LTE architectures, will be beneficial in terms of QoE estimation to prevent degradation, mainly thanks to its computing capabilities at the edge.

- On the AR/VR side, papers point out the need for a novel architecture [306], [368], [369] able to manage resources to trade performance, communications, or computing capabilities, taking into consideration the highly demanding AR/VR requirements, against the however limited MEC resources [307]–[310]. Recent papers could also leverage novel technologies such as terahertz frequency bands [318] or ML techniques such as DRL [321].

- Finally, some works start to focus on innovative but futuristic use cases such as XR and the Metaverse. In particular, papers explore using advanced techniques such as DRL or NOMA to support XR with a MEC system [322], [323], [324], while for the Metaverse, researchers investigate the creation of digital twin models [327], [370] or consider other aspects such as MEC-assisted users localization [328] and blockchains applied to Metaverse [371]. For interested readers, two interesting surveys have been recently published [329], [330].

There are several open research challenges:

- Regarding video streaming, only a few works addressed the **live** case, which imposes tighter requirements than classic video streaming. Live streaming websites such as *Twitch* and live video conferencing are becoming more and more important for the everyday user, especially in alert circumstances like the one generated by the Covid-19 pandemic, hence it would be interesting to dig more into how to improve the overall QoE, leveraging the MEC concept.

- While some works propose to use ML to forecast channel quality ([304]), the possibility of deploying an intelligent MEC node between the end users and a remote cloud server has not been fully evaluated yet. ML can help with smart caching, forecasting the video streaming load according to traffic patterns, and smart transcoding, among others, and therefore it will be useful in resource-constrained scenarios.

- Most of the available testbeds use LTE. While 5G is continuing its rollout in several countries around the world, some works appeared that use 5G testbed and MEC for supporting video streaming [392], [393]. However, it would be interesting to see more work relying on real 5G infrastructures.

- On the AR/VR side, many works focus on the highly stringent requirements and performance tradeoffs ([13], [308], [310], among the others), questioning whether edge/MEC

solutions would be a possible enabler. The answer is still unclear: while it is undoubted that for a fully interconnected VR, the road is still long, for baseline AR/VR, the MEC is however helpful for some task offloading, transcoding, and caching functionalities. However, current MEC solutions are quite limiting, also because MEC resources should be shared among different tenants, not necessarily belonging to the same vertical. In fact, for latency reasons, task processing delays caused by high AR/VR task demands might be still a relevant bottleneck for MEC and AR/VR applications. Therefore, tradeoffs between the edge computing infrastructure and VR devices should be further evaluated [368] (see Section 2.2 for further considerations). In the introduction chapter 1, we also focus on the energy aspects of AR/VR, showing that these use cases are computational and therefore energy energy-hungry. Most of the work done in the AR energy-efficiency domain focuses on increasing the energy efficiency of devices or base stations (or both at the same time). However, as mentioned in Section 2.1.2 the **Green MEC** paradigm could open novel possibilities for researchers to create a joint end-to-end energy efficiency scenario that jointly considers devices, base stations, and MEC.

- Another important new sector is **cloud gaming**. While existing solutions are somehow limited so far (*Nvidia Geforce Now*) due to the stringent requirements of gaming streaming (for instance, bandwidth requirements range from 10 Mbps for 1080p to a minimum of 35 Mbps for 4K[8]), 5G and MEC proximity deployments to end-users will surely help this paradigm to grow in terms of introducing newly available bands and offering smaller latency. This would open new possibilities to research (and to markets). One research challenge consists in enabling cloud gaming applications to leverage several access technologies at the same time to increase the overall QoS and QoE. However, at the time of this thesis, many cloud gaming services have closed for different reasons, with the biggest being Google Stadia. This remarks how the presence of MEC could be an opportunity for cloud gaming services providers to leverage added computing capabilities in the network, also thanks to the growing presence of dedicated hardware (e.g., GPUs).

Manufacturing: Another vertical that would benefit from the MEC presence is Manufacturing. Indeed, IIoT devices require low latency communications, high bandwidth, and computing capabilities, reliability, and security and at the moment only edge computing can satisfy all these requirements at the same time [394]. Reviewing the literature, we discover that:

- Most of the papers show the need to design a dedicated multi-level edge infrastructure for supporting smart factories, considering different constraints such as big data processing ([331], [333]), resource scheduling strategies ([332]) and reliability [334]. Compared to cloud solutions for manufacturing and smart factories in particular, a MEC infrastructure will decrease the computing latency and energy consumption by up to 40%.

- Other works show the need to make the MEC reliable for IIoT, to prevent deadlocks [335], and highlight how offloading to MEC needs to be made based on manufacturing task deadlines [336].

[8]https://www.forbes.com/sites/tiriasresearch/2020/02/04/nvidia-launches-affordable-geforce-now-cloud-gaming-service/#773ef8e1588b

- Recent papers also consider novel techniques such as blockchain [345], ML [339] or network slicing [375] while others start considering a ETSI-MEC compliant architecture standardization [342][343][341].

Many challenges remain open:

- The 5G ACIA has provided several useful insights on 5G deployment in smart factories [395], [396]. Focusing on the many MEC-related challenges in a smart factory, for instance, the MEC should be able to address a heterogeneous scenario consisting of several IIoT devices, each one with different demands and requirements. As an example, the MEC should support at the same time motion control devices (requiring a latency of $<$ 1ms), mobile robots (latency of 10-100 ms), and traffic for human-machine interaction (for instance through VR devices). Hence, it would be of fundamental importance to study (*i*) how the MEC could provide and manage at the same time different QoS constraints and (*ii*) its resilience when dealing with variable data traffic (such as bursts). A possible solution could be offered by leveraging network slicing for differentiating several slices according to the QoS required. Finally, while some work focusing on the manufacturing vertical has recently appeared (e.g., [397] and [398]) with some proposing also a ETSI-MEC compliant architecture, it is still interesting to evaluate how this vertical can benefit from a general edge computing standardization process.

- Also in this case, ML could be useful to solve some issues such as the ones related to the allocation of MEC resources. Furthermore, researchers could exploit the **Green MEC** paradigm to propose energy-efficient solutions for smart factories.

- Another very important aspect of MEC applications for manufacturing is security. The latter is fundamental for keeping IIoT data integrity. Otherwise, attackers might induce machine failure or product quality issues. Data confidentiality is also key because industry secrets must be protected. Security and **safety** in smart factories are very much tied since security breaches might cause malfunctioning of production lines and products, which could potentially harm workers as well as customers. A survey on the most common security attacks in NFV and 5G systems can be found in [199]. **Reliability** is a complementary aspect, since IIoT needs >99,999% of successfully transmitted packets. For instance, this can be achieved by deploying several MEC servers to create redundancy of resources (like in cloud datacenters) and/or a communications-wise management system, able to avoid extensive packet collisions. However, these solutions must also be **cost-efficient**.

- 5G ACIA also suggests exploring possibilities to converge together many communications technologies (D2D, Wi-Fi, antenna AP, sensors, RFID) to avoid wireless congested scenarios. As a possible solution, they propose to converge MEC and 5G with the Time-Sensitive Networking (TSN) framework, whose goal is to deliver deterministic services via IEEE 802 networks for wired industrial Ethernet solutions.

- Next research steps should also consider the novel architectures proposed by 5G ACIA and evaluate their solutions with **real data-driven** traces, to study MEC in real case scenarios. Furthermore, Digital Twins will become an important use case for 6G networks and it seems

that will be extremely beneficial for smart factories. Therefore the next question is how MEC could support digital twins (in general but also related to smart factories vertical.

eHealthcare: The advent of IoT devices is changing also the healthcare system, which now is becoming *smarter*. While it is true that it somehow overlaps with IoT, eHealthcare systems present unique features that can be exploited to design an effective MEC system. Vice versa, the MEC can be exploited to deliver unprecedented life-saver technologies. For instance, every patient might have his/her data processed independently and securely, and health alarms might be triggered reliably, avoiding privacy intrusions and false alarms, which means that the MEC should be thought as a secure and robust system. In turn, the presence of computing resources at the edge would help in the development of more sophisticated health machinery, which includes the support for remotely-driven surgery (e.g., tactile-Internet-based tele-surgery systems). By overviewing state-of-the-art works, we notice that:

- Most of the papers have identified MEC potentials for data pre-processing, to avoid sending too much sensible data over centralized clouds and to decrease sensible delays, up to 50%, compared to cloud-based networked healthcare systems ([346], [347], [350]–[352]).
- [349] showed that blockchain can be also exploited to increase the security level with eHealth devices while, with the same goal, [377] proposes to manage healthcare data in a decentralized manner thanks to MEC nodes' presence. Finally, [348] showed how edge resources could monitor a patient's health and be allocated in case of emergencies.
- Recent papers leverage data or scenarios from the COVID-19 pandemic [354], [355] or they focus on different problems such as task offloading in an eHealthcare scenario [353], [359], selecting the best computing model (if local or MEC) [360], leveraging D2D [356] or blockchains [358]. For interested readers, the authors in [378] delineate future research efforts for the eHealthcare vertical in 6G networks.

Many open research challenges are still open:

- As for the previous vertical, also eHealtchcare can leverage ML techniques. Indeed, due to the presence of edge computing resources, ML algorithms can be trained and applied to, e.g., quickly detect symptoms of diseases from images, therefore helping doctors in their diagnoses. In particular, federated learning seems a promising paradigm for privacy-related issues, since it allows us to maintain the data locally in multiple decentralized edge devices. **Blockchain** could also be used to add protection to personal data from malicious attacks and to make *auditable* the logs reporting the operation of the health staff, as well as the actions of patients. This might help to ensure that good practices are followed and would allow for to identification of conduct responsibilities in case of health issues.
- Moreover, it is interesting to notice that security and privacy are of fundamental importance in this vertical. However, adding advanced cryptography levels also increases computing overhead for resource-constrained edge nodes. Therefore, careful tradeoffs between security, computing, and the use of communications resources should be evaluated. For an overview of possible security threats, please refer to [199].

- Future works also address users/patients' mobility, with all the challenges it brings (see Section 2.1.2 for an in-depth analysis of the mobility challenges).
- While some works provided experiments in a real healthcare infrastructure [352], with the advent of even more wearable devices in the next years, it would be interesting to propose more **system** oriented MEC-related works, resulting in or driven by real data traces.

Many open challenges have to do with security/privacy aspects and machine learning. Here we do not analyze those aspects, because they have not been studied in light of MEC and edge-computing-related deployments. However, the interested reader could find more on those aspects in recently appeared surveys, e.g., [199], [201], [399], [400].

2.2. A smart metropolitan example

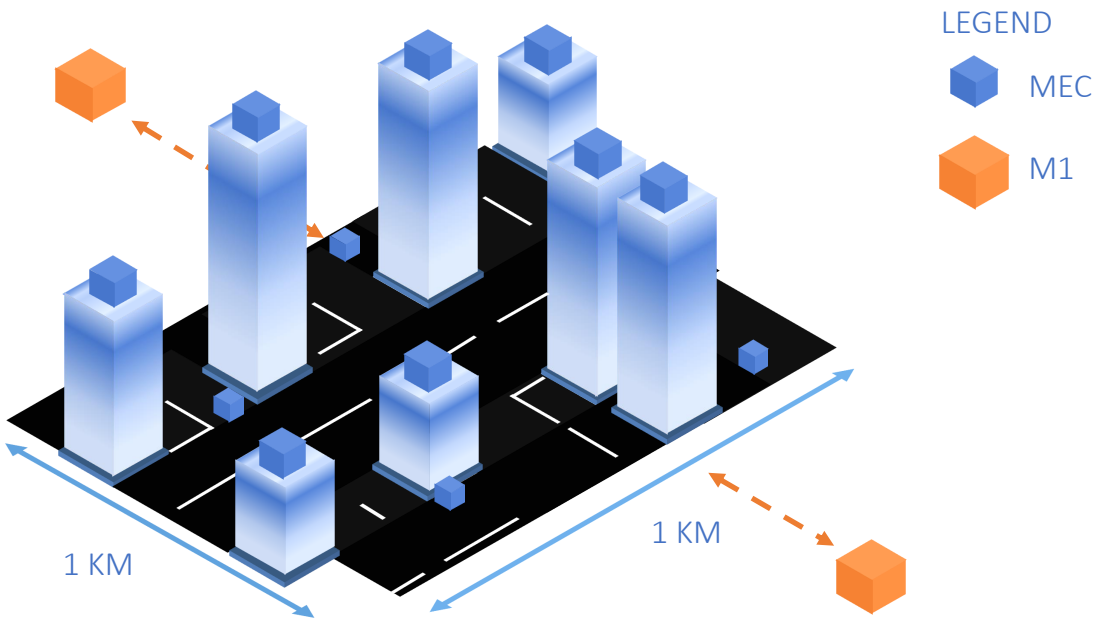


Figure 2.7: MEC deployment scenario in a smart city district.

In this section we highlight the features of the MEC deployment in a smart metropolitan environment, tackling the QoE requirements of citizens and workers, and the possible infrastructure bottlenecks, considering several verticals all together and a massive user presence.

The presence of connected devices is enhancing the cities and factories into smart entities with increasingly richer capabilities, evolving the concept from smart cities into a wider smart *metropolitan* area, which goes beyond the city itself since it includes a mix of areas where people leave and work, and also where services are produced and manufacturing happens. This allows for new communication and computing scenarios, e.g., for the interaction between (autonomous) vehicles and pedestrians, the dynamic management of electrical resources, and of AR/VR applications. To make these and other applications happen, it is of fundamental importance to guarantee the promised high data rates, high compute power, and low latency that came with 5G systems.

The edge computing paradigm is pivotal in this framework, and the MEC could be a key technology enabler.

Fig. 2.7 shows a district in a smart metropolitan area. The cellular network covers an area of one square kilometer and consists of twelve 5G antennas deployed near a corresponding MEC host, as described in [118]. According to the topology of access and transport networks proposed in [401], which is based on ITU recommendations [402], six 5G antennas (hence in our case six MEC hosts) are grouped under a single M1 access node, which is placed at an average distance of 10-20 km from the MEC hosts. Hence for each square kilometer, there will be two M1 dedicated nodes. Inside most of the biggest European cities, the M1 access node would hence be placed outside of downtown. Every group of six M1 access nodes is connected to an M2 node, typically located 80-100 km from the M1 node. However, here we do not consider M2 nodes and higher concentration nodes, whose distance from the user makes the propagation delay non-negligible. According to a recent (2024) report [403], nowadays in major European metropolitan areas the adoption of edge nodes is still at the beginning, with only a dozen edge nodes deployed throughout all of Europe. However, the same report shows that numbers are steadily increasing year by year, therefore likely reaching the number in the scenario presented in this section.

2.2.1. Network capabilities and use cases

Table 2.5 shows the values for network capabilities and requirements of MEC hosts, as suggested by 3GPP [404], and the corresponding values for computational capabilities, taken from [308]. In 1 km^2 the backhaul will offer a downlink (DL) capacity from the core of the network of 750 Gbps, distributed over two M1 nodes. The uplink (UL) will be more than 125 Gbps per square kilometer. Every MEC host can use, on average, at least 62.5 Gbps in DL and 10.41 Gbps in UL. With six gNBs per M1 node, these numbers correspond to the backhaul capacity of each gNB. These values are much higher than what can be offered by existing access network technologies, which therefore introduce a bottleneck for what concerns the actual speed observed by the users. For instance, the new standard for wireless communications 802.11ac will achieve a maximum throughput of 1.3 Gbps while the new 5G NR will achieve throughput up to a few Gbps [405].

Computational resources offered within the considered district are also quite powerful: for a MEC node located next to a gNB, it is possible to deploy a few servers (e.g., 16 servers), each with a few cores and GHz CPU rates (e.g., four cores at 3.4 GHz). For a MEC located on the M1 node, the number of servers can grow much higher, e.g., 256 [308].

In the example portrayed in Fig. 2.7, MEC resources are exploited only by users located outside buildings, without considering indoor hot-spots [404]. We consider a highly dense metropolitan area, with up to 25 000 users connected in square kilometer [404], which is the order of magnitude of the population density in the biggest European capitals. In particular, we build an example based on four representative use cases: (*i*) vehicle collision warnings with Cooperative Awareness Message (CAM) and Decentralized Environmental Notification Message (DENM) messages, (*ii*) video streaming and broadcasting, (*iii*) smart factories and (*iv*) VR/AR. Table 2.6 summarizes the per-use-case requirements, taken from [206], [406], [232], [407], [395], [334] and [367]. In our example, for the sake of simplicity, we assume that each user generates one task at a time for each request.

Table 2.5: Network capabilities and computational resources

	DL bandwidth	**UL bandwidth**	**Compute power** (machines × cores × CPU speed)
MEC host at gNB site	62.5 Gbps	10.41 Gbps	16 × 4 × 3.4 GHz
MEC at M1 access node	> 375 Gbps	> 62.5 Gbps	256 × 4 × 3.4 GHz

Table 2.6: Per-use-case requirements

Use case	**DL bandwidth**	**UL bandwidth**	**RTT**	**Compute power**
Vehicle collisions warning	4 kbps	70 kbps	10 – 100 ms	up to 43×10^6 cycles/task, minimum of 217600 tasks/s
Video streaming	70 Mbps	25 Mbps	10 ms	up to 1×10^9 cycles/task, minimum of 2176 tasks/s
Smart Factories	> 1 Mbps	> 1 Mbps	1 – 100 ms	up to 1.936×10^9 cycles/task, minimum of 114 tasks/s
AR/VR	4K 2D 100 Mbps 24K 3D 2 – 5 Gbps	6, 45 Mbps	30 ms 10 ms	up to 40×10^9 cycles/task, minimum of 6 tasks/s

Thanks to new connectivity possibilities, nowadays it is possible to improve road safety by leveraging CAM and DENM messages delivered from or to a vehicle, with collision avoidance algorithms processed at MEC nodes. The goal of these messages is to check if a collision can eventually happen and, in case, to warn nearby vehicles. Partially due to the small payload of messages, DL and UL minimum requirements for successful deliveries of CAM and DENM messages are quite small: 4 kbps for DL and 70 kbps for UL, while latency should range between 10 ms and a maximum of 100 ms [406], which corresponds to the generation rate of CAM messages (10 Hz) [206]. Instead, video streaming imposes more stringent and powerful requirements per service request: 70 Mbps for DL, 25 Mbps for UL, and a maximum latency of 10 ms [232]. However, the considered bandwidth requirements have been taken from measures from video broadcasting case scenarios (such as video sharing during a concert or sport live event inside a crowded stadium) and therefore, depending on the actual video streaming, requirements may vary.

Smart factory requirements vary as well, depending on the use case. For instance, controlling mobile robots needs at least a data rate of 10 Mbps and a latency of 10-100 ms, while motion control devices (such as packing machines) require less bandwidth (at least 1 Mbps) but a stricter latency (1 ms) [395].

The AR/VR use case imposes very stringent requirements: for a basic experience with 4K resolution of 2D videos, according to [367] [307], the DL bandwidth needed is 100 Mbps with an RTT of <30 ms, whilst for a full immersive experience (24K and 3D) the requirements go up to 2 – 5 Gbps of DL bandwidth, with less than 10 ms of round-trip latency [367]. A possible UL value for AR/VR applications is 6, 45 Mbps [408].

2.2.2. Bottlenecks and scalability

Taking a look at bandwidth capabilities, it is already possible to draw several conclusions: for instance, in the worst-case scenario, with all 25 000 users connected at the same time, the bandwidth for a fully immersive AR/VR experience cannot be guaranteed at all by gNBs and backhaul. However, if up to 30% of the users are connected (7 500 users), the full backhaul can provide enough

bandwidth for the most basic AR/VR applications. Still, with 1.5 Gbps available for UL at the antenna, the single gNB cannot serve more than about 15 AR/VR users with the least acceptable quality, so no more than 180 users can be served by the 12 gNBs present in the district. While fulfilling the enormous requirements for a fully interconnected VR is still a utopia, as also highlighted in [308], other verticals instead could benefit from the MEC presence in terms of bandwidth, computing capabilities, and latency. For instance, considering the same capabilities, the infrastructure can serve 22 video streamers per gNB, and a total of 264 users. In the other considered cases, the numbers grow very much. Indeed, a single gNB supports up to 1 500 IIoT devices in UL, enough for the average number of devices envisioned for a smart factory (according to [395], the number of IIoT devices could range from 2 up to 10 000 per km^2).

Furthermore, for vehicle collision warnings, the numbers are even higher: more than 375 000 parallel communication sessions are supported between the infrastructure and vehicles! Anyway, it is important to highlight that for smart factories and vehicle communications, MEC improved bandwidth capabilities are not as important as maintaining a reliability of 99.999%, otherwise, catastrophic situations could occur (such as collisions between vehicles or IIoT malfunctioning devices, with enormous economic damages for factories).

As a remark, real case scenarios are much more complex: gNBs and MEC nodes should be able to sustain *several* verticals at the same time, therefore providing bandwidth resources for AR/VR but also for vehicle collision warnings and smart factories, etc. When bandwidth becomes the bottleneck, a solution might be offered by deploying several gNBs per MEC node, although it would incur further CAPEX/OPEX costs. Furthermore, we need to consider computational and latency limits as well.

Looking at the latency requirements, they should always be guaranteed apart from the processing tasks delay. Indeed, thanks to the dense antenna and MEC deployment in the considered area, propagation delays on air, copper, and/or fiber are negligible (below half a millisecond per 100 km [409]), while processing packet delay at a MEC host running at medium load is of the order of one μs [410]. However, congestion must be avoided, which boils down to under-utilizing links and MEC hosts. For instance, typical queueing and computing architectures do not experience the buildup of large queues if used below $65 - 75\%$ of their capacity, depending on the distribution of task arrivals [411]. It is, therefore, safe to count only on two-thirds or at most three-quarters of the transmission capacity in the deployment (the same holds for computational capacity). For instance, this means that, at least from a point of view of available transmission resources, one should not accommodate more than $\approx 5\,250$ AR/VR streams and 1 050 IIoT devices.

Moreover, the numbers reported above need to be modified in case the computing power becomes the bottleneck. Specifically, focusing on computational resources, Figure 2.8 shows the maximum number of users U that can be served at the same time as a function of the processing cycles required to serve a task. We obtained these values by dividing the total computing power (CPU cycles per second) C of a MEC node by the ratio between the processing cycles (P_C) required for a task and the task deadline D (i.e., the number of CPU cycles per second required to serve a task):

$$U = \frac{C}{P_C} D. \tag{2.1}$$

In this example, we consider various numbers of servers, processing cycles, and task deadlines.

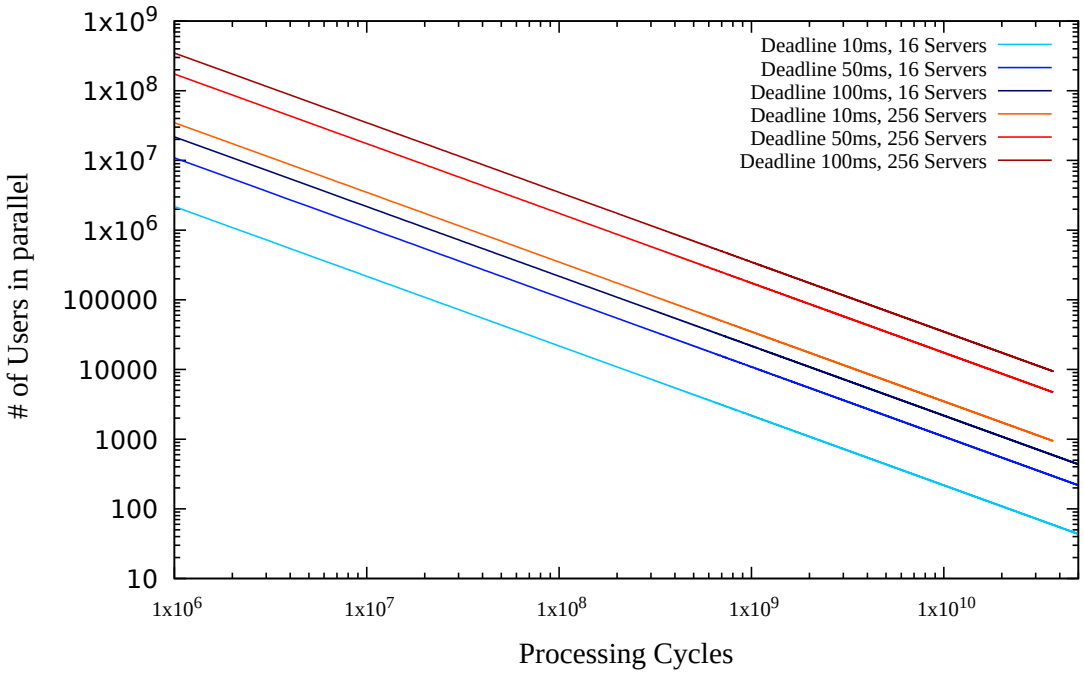


Figure 2.8: Number of users satisfied in parallel, according to different demands of processing cycles per task. Each user generates 1 task at a time for request.

Specifically, the deadlines reported in Fig. 2.8, i.e., 10 ms, 50 ms and 100 ms, indicate latency values that cannot be exceeded to provide optimal QoE ranging from video streaming services to vehicle collisions warnings, whereas server provisioning per MEC node consists in typical values of 16 or 256 servers. The heavier the computing tasks, the less the number of users that can be served at the same time, with an inversely proportional relation between the two quantities (which appears as linear in the log-log scale used in the figure). For instance, by exploiting light computing services, it is possible to serve more than 100 000 users respecting the deadline of 10 ms. It is the case of vehicle collision warnings, where short message size is mandatory to achieve faster information spreading across the vehicles. Further, it is possible to notice that both bandwidth and computing capabilities do not represent a clear bottleneck for this service, which therefore depends on the proximity of MEC resources [207] for latency issues. Instead, video streaming requires higher computational loads: considering a face recognition use case, a single task can require up to 1 billion cycles. So, taking into account a latency of $\approx$ 10 ms, the infrastructure can support a maximum of 3 000 users. The same happens for IIoT devices: according to [407] a critical task requires up to 1.93×10^9 cycles, hence $\approx$ 1 200 devices can be satisfied at the same time, a scenario one order of magnitude higher than the one described by 5G ACIA [395]. Finally, for a fully interconnected experience, AR/VR hits computation limits before bandwidth ones, and only 100 users can be served in the smart city district of our example in the case of 37 billion cycles per task for massive VR applications (as highlighted in [308]).

Summarizing, apart from the vehicle use case, all other considered cases show important limitations to support a massive presence of users (the scenario evaluated considered up to 25 000 people) in terms of computing capabilities. Furthermore, it is important to highlight two main aspects: (*i*) proposed computing capabilities are an over-provisioning exercise for very edge nodes (for instance a Nokia edge datacenter supports up to five servers) (*ii*) to avoid uncontrollable re-

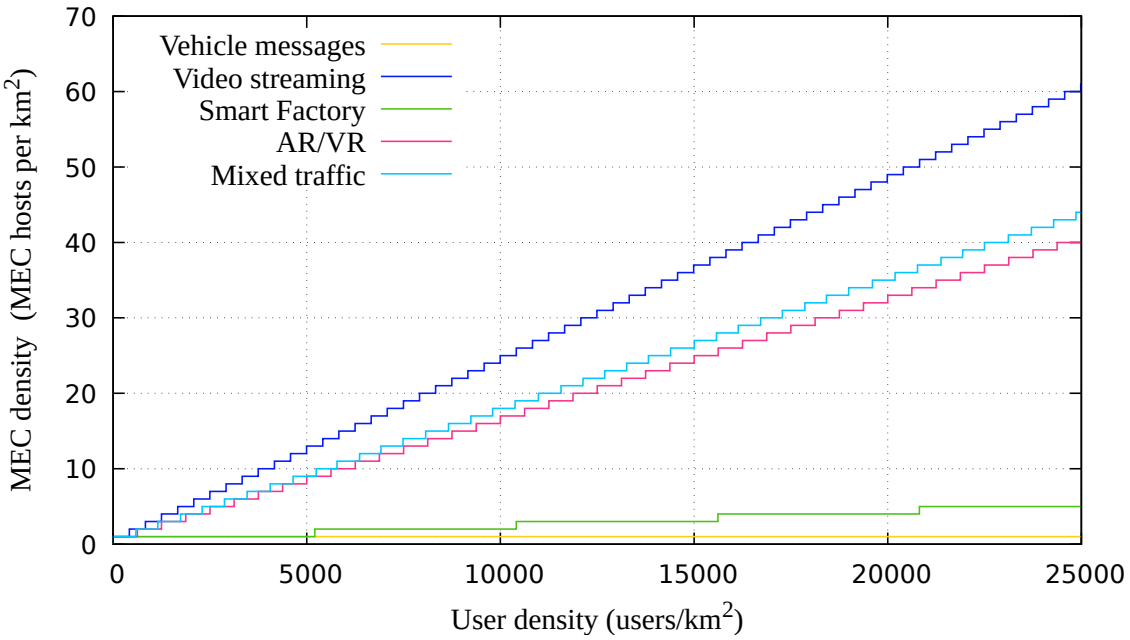


Figure 2.9: Density of MEC hosts required in different use cases, according to bandwidth requirements (refer to table 2.5 and 2.6 for parameters).

sponse times depending on the distribution of jobs arrival [411], MEC host capacity should not be exploited more than 65 – 70%. This means that, compared with the numbers described in Figure 2.8, for the same amount of processing cycles, servers should be used to serve no more than 65 – 70% of the nominal capacity, in terms of number of users. Therefore, if we consider a reduction of 50% of the server capabilities, which are then used only for the 65 – 70% of their full capacity, the numbers are quite different. For instance, now video streaming is supported for up to 1 050 users in parallel on the same MEC host, which is still more than what can be served with the available bandwidth. However, in the smart factory use case, up to 140 IIoT can be supported, and for the fully interconnected VR case, the MEC node can serve only 25 AR/VR devices. These numbers, compared to the ones obtained by considering the bandwidth, show that computation power can soon become the bottleneck. Furthermore, it is important to highlight that computing resources could be also shared among different slices. Therefore, as for the bandwidth capabilities case, fewer computing resources could be available causing a reduction of served users or an increase in processing delays [100].

The number of tasks per second supported by one MEC host varies depending on the number of users connected at the same time and it is inversely proportional with the processing cycles required. Table 2.6 shows the values in the worst-case scenario, when tasks require more processing cycles: for vehicle collision warnings, it is possible to sustain a minimum of 217 600 tasks/s, while the number drops to 2 176 tasks/s for the video streaming case. Furthermore, one MEC host supports up to 114 tasks/s in the smart factory case and 6 tasks/s in the AR/VR case. As for the previous cases, if we considered a reduction of 50% in MEC servers' capacity, and considering that they can serve only for the 65 – 70% of their nominal capacity, supported tasks/s would decrease accordingly (e.g., 40 tasks/s for a smart factory and a minimum of 4 tasks/s for AR/VR).

2.2.3. Required density of MEC hosts and its cost

Now we want to consider how many MEC hosts are required to support a given number of users connected simultaneously, considering different use cases for bandwidth constraints or computing power limits, and evaluate the associated cost (Fig. 2.11).

Fig. 2.9 shows the density of MEC hosts needed as a function of users density, according to the **bandwidth** requirements of different verticals, which are listed in Table 2.6. Here we considered one MEC host per gNB site. Firstly, it is important to notice the behavior of the warning collision messages use case: the bandwidth required for both DL and UL is so small that only a single MEC host can sustain the full range of user densities considered here (up to 25 000 users/km^2). Instead, for all the other use cases, the curve of required MEC hosts has a staircase shape. While the smart factory use case needs six MEC to sustain up to 25 000 users in a square km, therefore remaining under the threshold of 12 MEC hosts per square km proposed earlier, all the other cases (video and AR/VR and mixed traffic) require extra deployments of MEC hosts. If 25 000 users connect at the same time in a square km area, and everyone leverages on AR/VR services, at least 41 MEC hosts are needed in that area to guarantee enough bandwidth for both DL and UL, while 62 MEC hosts are needed for the video use case. The curve labeled as *Mixed traffic* represents a scenario in which a mix of all four use cases is present. Specifically, we design the mix of traffic according to Cisco[9] and Ericsson[10] traffic forecasts for the following few years: 70% video traffic, 15% car traffic, 10% smart factory and 5% AR/VR. For 25 000 mixed users/km^2, more than 40 MEC hosts per square km are required (i.e., more than the 12 proposed earlier). In all of our considered cases, UL imposes tighter constraints, so that the bandwidth bottleneck is imposed by the aggregate UL traffic.

Furthermore, we study the required density of MEC hosts according to **computing power** needs of the users. In Fig. 2.10, we consider the presence of the MEC hosts at gNB sites as well as M1 nodes, with a ratio of one M1 node for every six MEC hosts. To avoid unrealistic deployment scenarios, we limited the deployment of new MEC hosts up to 96 (hence $\approx$ one MEC hosts per 100 m^2). This justifies the curves ending before reaching the maximum population density considered (i.e., curves stop where the capacity of 96 MEC hosts per square km, and the associated M1 nodes, have been reached). In the figure, we notice that, while the vehicle warning messages use case again requires very low computing capabilities (only one MEC host per square km), the other cases behave differently. The AR/VR use case saturates the MEC computing capacity (up to 96 MEC hosts) within less than 2 000 served users. Instead, video streaming, smart factory, and the mixed traffic scenarios are all able to sustain a traffic of 25 000 users/km^2 or more. More specifically, up to 82 MEC hosts plus 16 M1 nodes are needed in the area of one square km to serve the mixed traffic case. The numbers go down to 60 MEC hosts and 12 M1 nodes for the smart factory case, and further down to 22 MEC hosts plus 6 M1 nodes for video streaming. It is possible to notice that, apart the the video streaming use case, all other cases require more MEC nodes to provide computing capacity than what they need for bandwidth. This shows that computing represents the real bottleneck in most of the cases.

[9]https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[10]https://www.ericsson.com/en/mobility-report/reports/november-2019/mobile-traffic-by-application-category

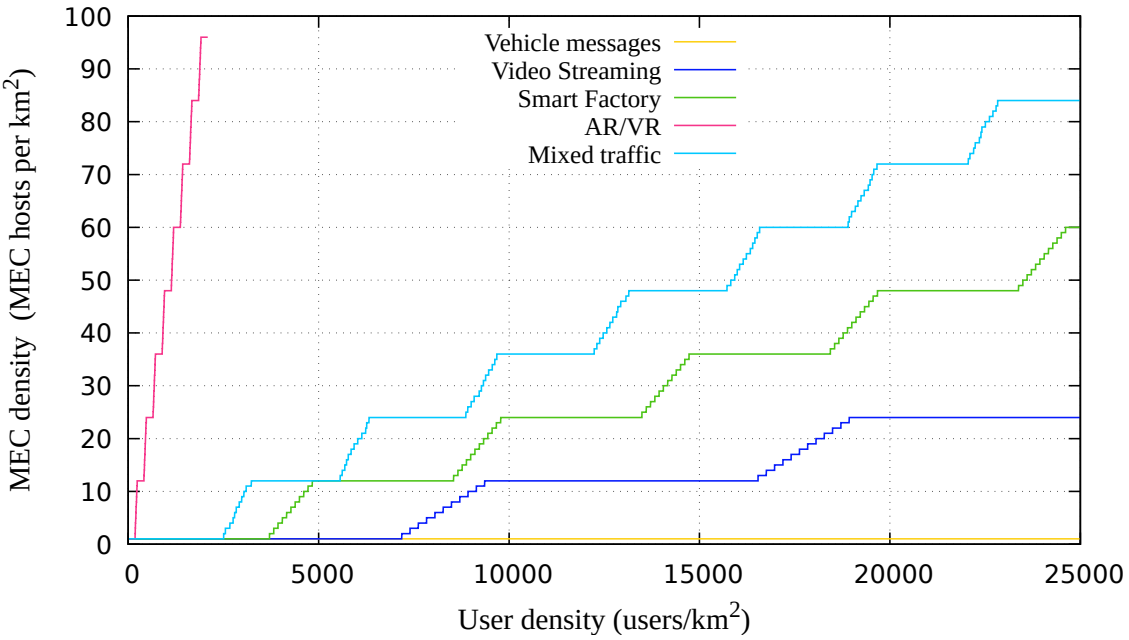


Figure 2.10: Density of MEC hosts which are required, with their associated M1 nodes, to serve different use cases based on computing power requirements (see Table 2.5 and 2.6 for parameters).

Fig. 2.11 shows the infrastructure CAPEX needed to deploy MEC nodes in a square km, as a function of user density. This time we consider both bandwidth and computing power requirements. In the figure, we consider a cost of ≈ 2 000 USD per deployed server plus other CAPEX costs such as new base stations deployments, civil works, and small cell equipment (≈ 94 000 USD) [35]. We also consider the cost of M1 nodes, for each of which we use the extracted CAPEX cost of deploying a 256-server datacenter (≈ 1.5 million USD) evaluated with AWS cost calculator[11].

From the figure, it is possible to see that, again, the cost to sustain the vehicle warning messages case remains steady, due to the low bandwidth and computing requirements. Video streaming and smart factories have the same long-term behavior: they can sustain the whole population while reaching a final cost of 25 million USD per km^2.

In the mixed traffic scenario, the cost to sustain as many as 25 000 users per square km is higher, summing up to 34.8 million USD per km^2. In the AR/VR case, we observe the highest costs (up to slightly more than 35.8 million USD with less than 2 000 users/km^2). However, it is interesting to notice how the AR/VR use case alone increases the infrastructure CAPEX costs, therefore giving a new design constraint to infrastructure providers. This behavior could be viewed especially in the mixed traffic scenario, where it contributes only 5% to the overall traffic.

Summary: We showed how in a smart metropolitan context both the bandwidth and the computing capabilities, even when quite powerful, require the deployment of new MEC nodes, exceeding therefore the threshold of 12 MEC nodes. Furthermore, we showed that especially the computing capabilities represent a clear bottleneck for the network infrastructure. This however

[11] https://awstcocalculator.com

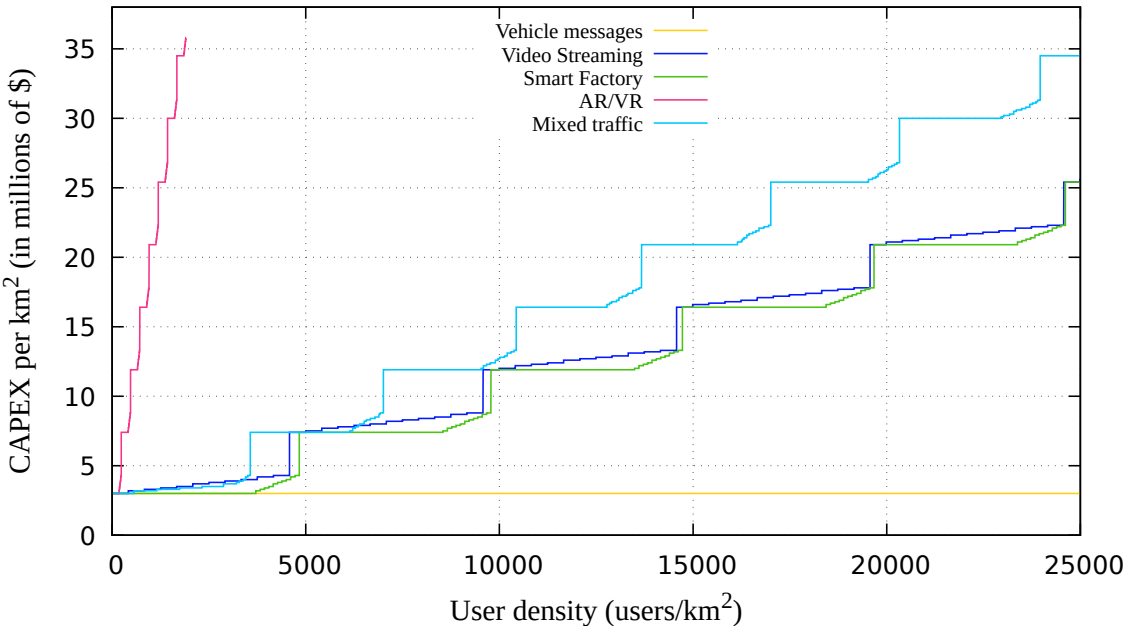


Figure 2.11: Infrastructure CAPEX cost per km^2.

does not mean that a heterogeneous smart metropolis is not possible: while advanced AR/VR is still much beyond the nowadays network capabilities, connected cars exchanging simple collision warning messages together with video streaming and smart factories might coexist together, placing a first step towards the path of a fully interconnected metropolitan area.

2.2.4. Open challenges

The analysis of network deployment in a smart metropolitan area highlights some lessons learned and points out some problems that need to be addressed: First, the computational capabilities of the MEC deployment should be carefully considered as a function of the expected verticals operated in the served area, since different verticals (e.g., video streaming, smart factories or AR/VR) have very different requirements [412]. Second, it seems very impractical, from a pure cost perspective, to scale up a typical 5G use case for a big crowd of devices. For instance, according to Intel[12], in the next future a single autonomous driving car will generate up to 4 terabyte of data per day, which would require either very powerful MEC hosts or very dense and expensive MEC deployments just to serve a few tens of cars per unit area. Third, future MEC host solutions should consider leveraging GPUs instead of CPUs for entertainment use cases such as gaming or AR/VR, and leveraging ML per forecasting task arrivals, allowing to allocate/scale resources in advance. In addition, they could leverage smart computation offloading in order to avoid unnecessary offload to MEC hosts.

In addition to the above points, our simple examples show that the MEC deployment in a urban district or a metropolitan area can require high densities, which incurs logistic problems and constraints, and hence requires carefully designed deployment plans which account for presence

[12]https://newsroom.intel.com/editorials/krzanich-the-future-of-automated-driving/

of natural or artificial obstacles while guaranteeing uniform reachability and access to bandwidth and computational resources. The use of renewables would be desirable, in accordance to recently proposed energy-awareness efforts, e.g., to follow the guidelines of the European Green Deal agenda[13] or of the Microsoft Green Computing initiative[14].

2.3. Summary and conclusions

In this Chapter, we discussed some general aspects of MEC and how they will shape the future of edge computing capabilities in cellular networks. More in detail, we tackled the ETSI efforts in standardizing MEC, discussing at a general level what has been done by both SDOs and researchers and we showed how standardization processes are going to be important in the future networks, also considering the progress in other closely-related domains (such as O-RAN). We commented on the overall ETSI-MEC architecture and how it will merge with the ongoing rollout of 5G networks. Next, we tackled some of the most challenging aspects of the MEC provisioning. We focused on the difficulties in proposing and evaluating certain scenarios such as jobs/tasks offloading, migration of edge resources, and the (flexible) deployment of MEC nodes in cellular networks. We overviewed several different techniques or technologies, such as ML, optimization techniques, stateful/stateless migration, containers, and VM, commenting on benefits, drawbacks, and possible future research challenges. In particular, we understood that due to the scarce presence of computing resources at the edge, the migration of resources will play an important role in the future for allowing the sustainability of edge infrastructures, while also targeting the stringent QoS and QoE of future use cases. Afterward, we showed how MEC could support vertical industries. We surveyed recent papers from the literature, highlighting several approaches and techniques tackled by researchers while also commenting on some possible future works and scenarios. We showed how several verticals have not been fully yet, leaving room for further studies. Finally, we considered a smart metropolitan scenario, exposing several constraints (e.g., bandwidth, computing, and economics) and limitations on supporting several verticals at the same in a future smart metropolitan context. In the next chapters of the thesis, we present two different novel scenarios stemmed from the above extended literature review and we provide two efficient solutions, leveraging different techniques. More in particular, our two novel scenarios are edge gaming and MAR, which will have an important impact on future cellular networks. In the previous chapters, we showed that while MAR is an object of ongoing study between several researchers, edge gaming is still in its initial phase but it is gaining traction thanks to the more and more powerful computing (i.e., GPU) capabilities. However, both use cases are computing-hungry and have stringent QoS constraints and therefore the presence of MEC, as commented in Table 2.4, is mandatory to support them. At the same time, sustainability (in terms of carbon footprint) is going to be pivotal in 6G networks [11], [12] and the **Green MEC** paradigm, as highlighted earlier sections, has not been evaluated yet by the research community. Therefore, we consider the study of both scenarios (edge gaming and MAR) taking in mind a sustainable effort, with MEC nodes that could (partially) depend on the availability of renewable sources. Our main goal is to make the edge infrastructure *sustainable* for a network/edge operator, meaning that we want to leverage as much as possible the intermittent presence of renewable sources but also the infrastructure should be prof-

[13] https://ec.europa.eu/info/strategy/priorities-2019-2024/european-green-deal_en
[14] https://blogs.microsoft.com/green/

itable (i.e., the infrastructure should be able to serve as much paying users as possible, therefore increasing the overall revenues). These two goals (maximizing revenues while decreasing carbon footprint) are conflicting and some intelligent algorithms should be developed. One possible way to deal with this scenario is to allow the migration of already allocated resources (e.g., jobs/tasks) within close-edge servers, thus following the presence of renewable energy to decrease costs and also smartly leaving space for more arriving tasks (with increased revenues). In the thesis, we propose two different solutions to tackle these problems: one is an intelligent heuristic that allocates and migrates jobs according to the presence of available green energy. We show the robustness of this approach in different scenarios. Next, leveraging machine learning techniques, we propose a DRL-based solution, showing that DRL approaches achieve better performance when they have to deal with a proportional fairness structure (i.e., finding a balance between revenues and carbon footprint).

3. GREEN EDGE GAMING

In this Chapter, we tackle the problem of how to support gaming at the edge of the cellular network. According to what we discussed in the previous Chapter 2, the edge gaming paradigm represents a promising concept that has not been fully evaluated yet. Indeed, the reduced latency and higher bandwidth that the edge enjoys with respect to cloud-based solutions implies that transferring cloud-based games to the edge could be a premium service for end-users. At the same time, edge servers have scarce computing resources compared to cloud ones and for sustainability reasons, they could depend on intermittent renewable energy sources. Therefore, it is critical to efficiently allocate computing resources in such a constrained scenario (i.e., the edge network), especially when considering computing demanding use cases, like gaming sessions. As mentioned in earlier chapters, the flexible provisioning given by NFV allows for a dynamic allocation and migration of resources, helping network operators achieve their goals (e.g., maximizing revenues and decreasing carbon footprint). The goal of this chapter is to design a scheme compatible with MEC and network slicing principles of 5G and beyond, maximizing the utility of a service/infrastructure provider with time-varying edge node capacities due to access to intermittent renewable energy. We formulate a *multi*-dimensional integer linear programming problem, proving that it is NP-hard in the strong sense. We mentioned as an important aspect of this dynamic scenario the possibility of migrating resources across different nodes. Therefore, we propose an efficient heuristic, `GREENING`, which considers the allocation of gaming sessions and their migration according to the presence of *available* green energy. For the mentioned scenario, we analyze a wide variety of realistic configurations at the edge, studying how the performance depends on *(i)* whether the games have a static or dynamic workload, *(ii)* the distribution of renewable energy through nodes and time, and *(iii)* the topology of the edge network. Through simulations, we show that our heuristic achieves performance close to that achieved by solving the NP-hard optimization problem, except with extremely lower complexity, and performs up to 25% better than state-of-the-art algorithms.

The rest of this Chapter is organized as follows. We give an overview of the importance of edge gaming in Section 3.1, while also defining the novel concept of *green* edge gaming. Afterward, we provide the system model in Section 3.2. In Section 3.3, we formulate an instantaneous optimization problem, proving its NP-hardness and submodularity. In Section 3.4, we tackle the more general online problem of green game session allocation and we propose our efficient heuristic: `GREENING`. Section 3.5 highlights our main results and finally Section 3.6 summarizes the Chapter.

3.1. Background

Cloud gaming allows users to play the newest-generation games requiring only an internet connection and a screen (e.g., a TV screen, laptop, or mobile phone) by leveraging a cloud infrastructure. Games are located and processed in a cloud server, which streams the content to the end-user screen. Initial attempts were unsuccessful (e.g., OnLive) mainly due to the lack of infrastructure, but nowadays cloud gaming is having a second life. It is a growing market—reaching by 2023 a total revenue of eight billion dollars [16])—and many tech companies are launching cloud gaming

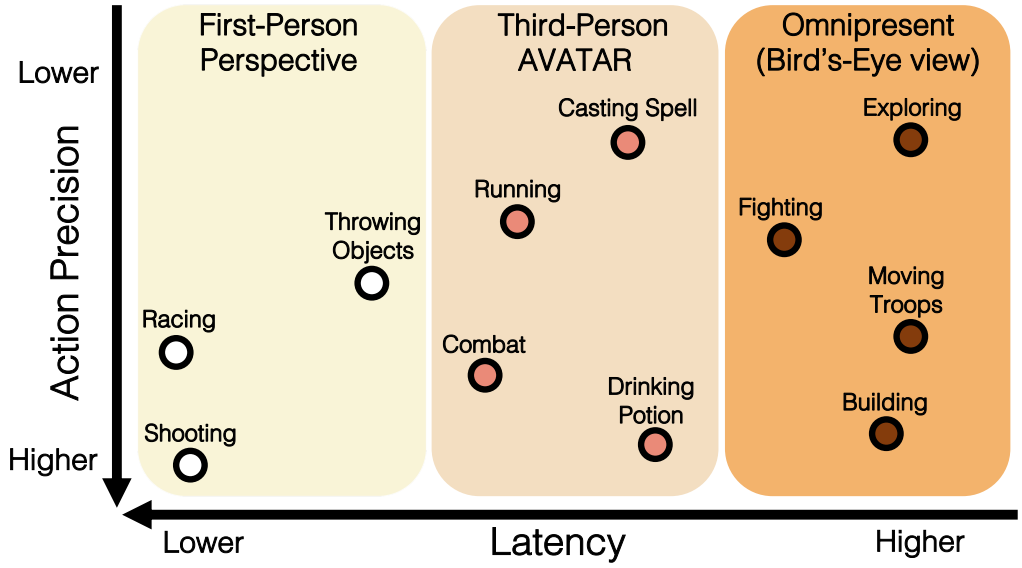


Figure 3.1: High level example of latency requirements for different game actions.

services within their network infrastructure, for example, NVidia with Geforce Now, and AWS with Amazon Luna, to name just a few. In 5G and beyond, MEC allows exploiting cloud gaming at the edge, developing the concept of *edge gaming*.

We can highlight three main benefits of edge gaming with respect to cloud gaming: (*i*) notably reduced latency, which makes the gaming experience immersive and interactive, with different nuances, as shown in Figure 3.1; edge gaming will enable in particular fast-paced games, where timing is fundamental—e.g., First Person Shooter (FPS)—and competitive online multiplayer gaming. (*ii*) At the same time as allowing for tighter latency requirements, allocating games at the edge will reduce network core congestion [28], therefore reducing the possibilities of packet losses while allowing higher video quality. Finally, (*iii*) the edge gaming paradigm could easily leverage the roll-out of new networking principles such as network slicing and new MEC standards defined by standardization bodies such as ETSI [9].

However, constrained edge resources are scarce and variable, and new techniques should be proposed to efficiently provision resources to meet several quality and cost constraints. In particular, it is expected that energy will become a crucial bottleneck for the deployment of this kind of systems [18], and there is a growing interest in making infrastructures more sustainable by leveraging renewable energies [413], [414]. Indeed, edge nodes and datacenters could be endowed with their own sources of renewable energy, which means that MEC nodes would have time-varying capabilities depending on the fluctuating energy resources, raising the possibility of migrating resources across several edge nodes if necessary. This is particularly true for gaming applications, as games have a highly dynamic behavior in terms of both workload and instantaneous resource requirements [14]. As a matter of example, we can imagine that the workload required for the same game differs when it renders a static scenario with respect to the case where the scenario quickly changes (e.g., where the character moves); such change of requirements may happen also if the frame rate has changed from 30 to 60 fps.

Games could exploit migration at the edge because migration delays are negligible in such a scenario due to the proximity of edge servers in residential areas [1] and new efficient migration techniques that can be exploited [61], [62], [415]. Hence, the possibility of migrating online gam-

ing servers in a few milliseconds would enable seamless game sessions and therefore an increased QoE for end-users even in a dynamic scenario with twofold variability—in game requirements and node capabilities. In such a (possibly unsteady) scenario, resource allocation and migration become one of the most challenging tasks.

Main contributions

Motivated by the aforementioned, we focus on the problem of resource allocation in a *sustainable edge-based online gaming* scenario, which we refer to as **green edge gaming**. In this scenario, we aim at maximizing the utility of the system taking into account revenue and costs of energy, deployment, and migration. Accordingly, we develop a smart allocation and migration algorithm for online game sessions under several realistic constraints.

The main contributions of this chapter are as follows.

- We develop the concept of green edge gaming with time-varying edge node capabilities due to the fluctuating availability of renewable energy. We show that this concept is compliant with ETSI MEC standardization and with modern networking principles such as network slicing.
- We argue that this concept could lead to a *premium* business scenario for which we formulate an accurate *multi*-dimensional linear integer programming problem, showing that it is NP-hard in the strong sense and sub-modular.
- We develop `GREENING`, an efficient online heuristic for game session allocation and migration which maximizes the utility by maximizing the use of renewable energy (green energy) instead of the one proceeding from polluting sources (brown energy).
- We study the proposed algorithm in realistic settings, where (*i*) the amount of available renewable energy is obtained from a database of real values for solar and wind energy generation whose average levels depend on the time of the day; (*ii*) maximum capabilities of edge nodes are taken from actual commercial equipment; (*iii*) resource requirements of game sessions can be either static (e.g., as an approximation for their maximum possible values) or dynamically change at a fast pace, in accordance with real measures of online gaming. We are the first to model and study the job's dynamics in a gaming scenario at the edge.
- We evaluate the proposed algorithm against several benchmarks, and we show they achieve near-optimal performance without high complexity. The results show that the proposal obtains values up to 25% better than state-of-the-art approaches.

Novelty of our work: To the best of our knowledge, the *green edge gaming* scenario, in which games are allocated in nodes with time-varying capacity and in the presence of both fluctuating energy and workloads and of several nearby edge nodes, has not yet been analyzed. In this scenario migration of tasks may play a significant role, since they can not only reduce costs and pollution, but also avoid congestion in the network core, with improved QoE for end-users.

Many works in the cloud gaming area have tackled QoS or QoE metrics (on delay and bandwidth especially), although they do not incorporate other aspects as storage, energy, or computation limitations, since these constraints are usually not challenging in a cloud infrastructure due to higher amount of resources. However, they are indeed very important for the edge infrastructure. In this Chapter, we attempt to consider all the types of resources that may become the bottleneck in the green edge gaming scenario, namely bandwidth, delay, storage, node computation capabilities, and energy available. Besides, we consider a twofold dynamic setting, where we consider that both the capacity of the nodes (due to variable amount of renewable energy) and the requirements of the jobs (due to the inherent varying nature of games specifications) are time-varying, which has not been evaluated before in edge/cloud gaming scenarios.

Most of previous works focused on a subset of the constraints here considered and the migration of tasks, if taken into account, was mainly based on user's mobility. We will show in this work that migrations have a fundamental role independently of user mobility. We assume that the migration of already-on-the-system jobs can be performed within nearby edge servers for two main reasons: first, to optimize the use of energy and other resources, but also to make space in the system for newly arrived jobs while respecting all the considered constraints. This is only possible in the edge context, since migrations cannot be considered in legacy cloud gaming contexts [24]. Finally, we are the first ones that study the jobs' dynamics in an edge gaming scenario, exposing how resources should be delicately allocated at the edge with jobs having dynamic workloads. In this scenario, migrations are necessary in order to maintain a high QoE for end-users.

3.2. System Model

Figure 3.2 shows a schematic of our reference scenario. We model a layered 5G edge network infrastructure [416] containing a set of edge game servers, with a set of network links connecting these servers between them and with the end-users. We denote the set of network links by $\mathcal{Z}$, and the size of this set as $Z = |\mathcal{Z}|$. The set of edge computing servers is denoted by $\mathcal{N}$ and it is composed of N servers. We consider two types of servers, which differ in their capabilities and proximity to end-users. Among the N nodes, B servers reside on *far-edge nodes*, each deployed at a base station (BS), and primarily meant to serve users of that BS, whereas the rest of the servers are each located at a different *M1 node*, placed in the edge/transport network where the traffic of multiple BSs converges [416]. M1 nodes have bigger capacities compared to edge nodes, in terms of computation, energy, and memory capabilities, although these capabilities must be shared across users belonging to multiple cells.

We distinguish two different types of energy powering the servers, whether it comes from renewable (green) and non-renewable (polluting) energy sources. Regarding green energy availability, we restrict ourselves to locally generated energy, and thus we only consider wind and solar as renewable sources, which have already been applied in edge computing contexts [417]. Non-renewable energy is always available at M1 nodes, while edge nodes might not have access to it. When a server can only access green energy, its computing capacity is proportional to the available green energy. In all cases, we assume that green energy can be used at no cost, but brown energy has a non-negligible cost.

The infrastructure is used to run online game sessions, each of which is referred to as a *job*.

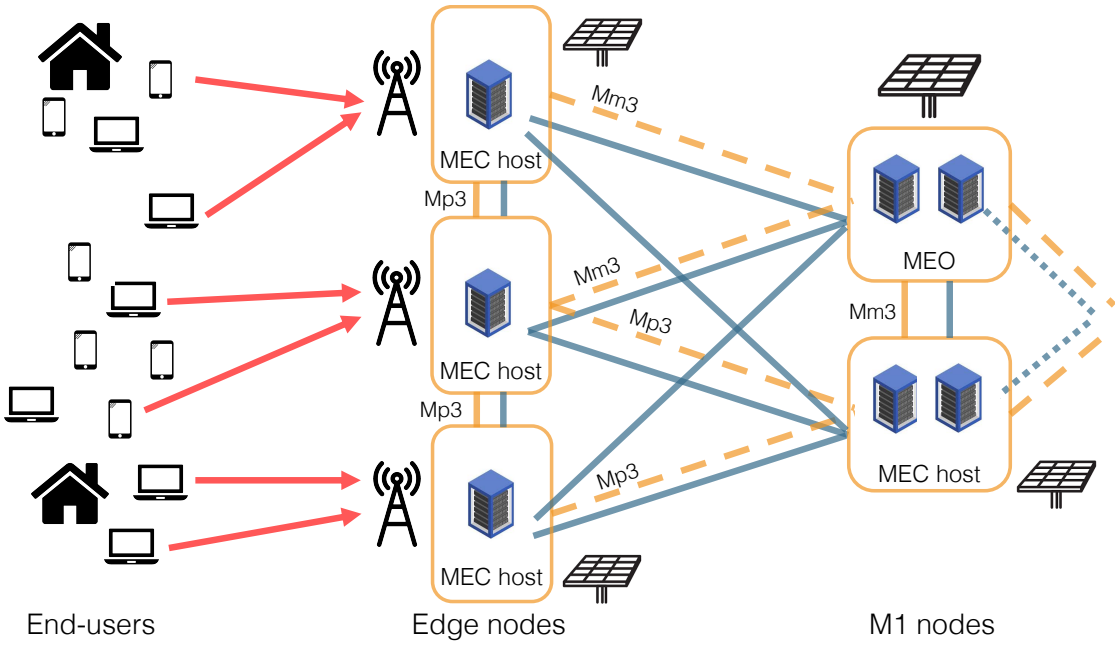


Figure 3.2: 5G Edge Infrastructure compliant with ETSI MEC.

The operation time is slotted, and we consider that the game sessions' requirements are random variables that may follow different distributions. The jobs arriving over time are modeled by a set $\mathcal{J}$. The operator accepts to process the job in exchange for a monetary payment, such that a certain job j provides a revenue R_j to the operator that includes many factors, e.g., user's fees, percentages of game purchases, advertising, etc. At the same time, jobs incur a cost of deployment, management, and processing which depends on the amount of computation, memory, and communication resources, the energy required, and the duration of the job, which are variable quantities that evolve. Furthermore, job interruption (due to a shortage of energy) or migrations also incur a certain non-negligible cost.

In the following, we explain in detail the time-slotted operation of the system, and the statistical model considered for energy fluctuations, jobs requirements, and nodes capabilities.

3.2.1. Resource allocation for Green Edge Gaming

The system operates in a time-slotted manner. We consider a centralized decision maker that is aware of the state (in terms of capability and load) of each server. At the beginning of each time slot, there exists a set of newly arrived job requests, as well as another set of jobs that are already being served. Furthermore, the amount of renewable energy at each node and the energy and computation requirements of each job may vary from one-time slot to the next. The system's task is to migrate ongoing jobs, interrupt them, and accept and allocate current requests to optimize the utility of the scenario. The optimization is applied each time slot in which something has changed, i.e., upon new jobs arrive, the available level of green energy changes, or job requirements change. Note that considering dynamic energy levels and workloads is challenging because past optimal job allocations might soon turn into non-optimal and call for reconfiguration at a frequent pace. We will analyze different settings in which the frequency of these changes varies.

The network does not know the future duration of each job, as game sessions have an unknown duration in nature. However, we assume that at each time slot, the decision maker knows the job requirements (bandwidth, delay, memory, computation, and energy) for the starting slot. This assumption can model a scenario where the network can estimate and/or predict with high enough precision the average consumption of a certain job based on the information available (type of game and device, previous values, etc.) and the considered optimization time slot is short enough, e.g., a few tens of seconds for a game whose computing and rendering power typically change significantly only upon significant changes of scene.

3.2.2. Energy fluctuation model

We consider that each edge node is equipped with on-site renewable energy sources. In particular, we consider that each edge node has installed a personal-use-size windmill and a one-square-meter solar panel. Both energy generators amount to a total maximum capacity of 1.5 kW, as per specifications of current commercial devices.[15]

Due to the unpredictability of wind/solar resources [50], we model the green energy behavior in a stochastic manner [418]. We make use of the dataset provided by a Belgian operator called Elia to create samples that match the trends and randomness of green energy generation in a real power grid. In particular, Elia provides weekly forecasts of both wind [419] and solar [420] energy generation in Belgium, with a granularity of 15 minutes, and we have used the data generated for the period from 21st to 27th of March 2022 for specific areas in Belgium.

In addition to the most probable forecast, the dataset provides confidence intervals. We will use such information to generate random realizations of energy forecasts that conform to daily changes in green energy availability. Figure 3.3 shows a weekly solar and wind power generation forecast together with confidence intervals, with data taken from the dataset made available by Elia. We can observe that the amount of available green energy varies considerably depending on the time of the day but also between different days. For simplicity, we do not consider the use of long-duration batteries at edge nodes, and therefore green energy is not stored from time slot to time slot. Another reason is that we are interested in leveraging as much as possible the presence of green energy. In other words, we are in a scenario where there are continuously arriving jobs and we want to maximize the use of the edge infrastructure to maximize the revenues, while at the same time using as much as possible all the available green energy.

3.2.3. Job monetization and cost

Each accepted job brings a revenue R_j to the operator, which may depend on the requirements of the job. At the same time, jobs require an operating cost that is proportional to such requirements. In particular, the total cost associated with job j, which is denoted by C_j, is composed of the cost of deployment, the cost of the non-renewable energy consumption, the possible cost from migrating the job, and the interruption cost.

[15] E.g., see *Tumo-Int* 1000*W Vertical Wind Turbine Generator* or *GONGJU* 1000*W Vertical Axis Wind Turbine Generator* for windmills, and *Jinko TIGER Pro 545W* or *Longi Hi-MO* 4 *455W* for solar panels.

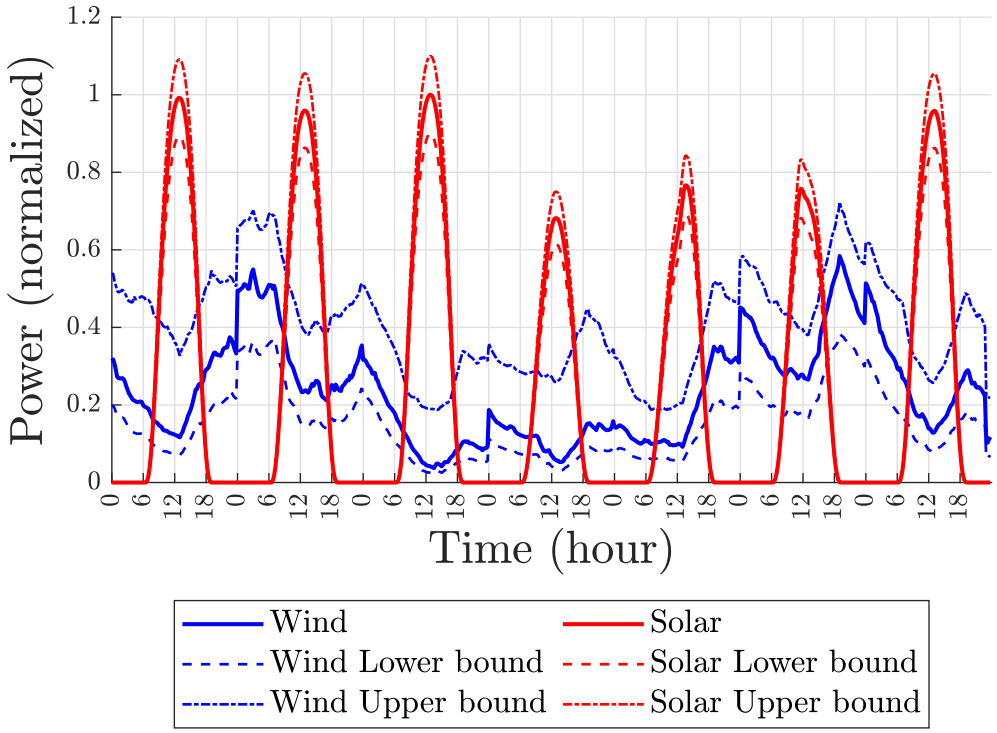


Figure 3.3: Weekly solar and wind power generation forecast provided by Elia for Flanders (wind data) and for a federal region of Belgium (solar data), from the 21st to the 27th of March, 2022. Values reported in the figure are normalized to the solar peak average expected on the third day (about 3 MW for the entire region to which the dataset applies). This forecast dataset was re-scaled to account for the fact that only a limited number of solar panels and a windmill can be mounted at an edge node, and used to produce the numerical results presented in Section 3.5.

a) **Deployment cost:** It represents the cost of instantiating and deploying resources to support the accepted jobs, and it is denoted by $C_j^{(d)}$.

b) **Energy cost:** This cost is proportional to the amount of polluting energy consumed by the job at each time unit, and hence is not constant over time. It is denoted by $C_j^{(b)}$.

c) **Migration cost:** The migration cost represents the induced operating cost derived from re-deploying, re-scheduling, and migrating resources among nodes within the edge network.

For simplicity, we do not consider migrations triggered by handovers as an optimization problem. We do so not only because that topic has been covered in other studies [61], but also because we are interested in the evaluation of interactive game sessions, which are typically several minutes long and are played at home or in a static environment [421].

d) **Interruption cost:** A job interruption leads to a loss of performance and the termination of the user experience. Consequently, the cost associated with interruptions ($C_j^{(p)}$) is considerably higher than that of migrations and can take out the revenue associated with that job, because of the *premium* nature of the user's subscriptions. Accordingly, jobs have to be scheduled immediately or rejected rather than queued. A job interruption can occur both when the availability of green energy decreases and/or allocated jobs change their workload, causing the server's computing or energy capacity to become insufficient for all running jobs, and migration cannot be enforced.

Note that our approach to revenue and cost values does not consider topological factors such as the distance between nodes. Those factors lead to negligible differences in the edge scenario (cf. [1]).

3.2.4. Game requirements model

Jobs originate from devices such as mobile phones, laptops, or smart TVs. Thus, considering a large potential number of users, we consider that jobs arrive according to a Poisson process and have a duration extracted from a Weibull distribution, which realistically models the duration of online game sessions [421].

Every job will need powerful dedicated resources to work smoothly, in particular for energy [18] and computing power. The jobs, which we recall that refer to online game sessions, must meet QoS requirements in terms of delay and bandwidth, and they are characterized by their requirements in terms of energy, memory, and computation consumption.

Let us describe separately these aspects.

a) **Delay:** The overall response delay is the total time between an end-user submits his/her commands and the time the corresponding game frame is displayed to the user [422].

Response delay (D_r) is composed of network delay (D_n), processing delay (D_p), game logic (D_g) and playout delay (D_o), i.e., $D_r = D_n + D_p + D_g + D_o$ [422]. The network delay is the round-trip-time (RTT), which depends on where the server is placed; the processing delay is the delay to encode/decode and packetize commands and frames (which could take from 5 ms to 100 ms, depending on many factors [62], [422]); game logic delay denotes the time required by the game software to process a user's command and render the next game frame that contains responses to the command. This delay strongly depends on the game, with a range from 5 ms [62] to 50 ms with cases reaching even 130 ms [422]. Finally, the playout delay is the time required for the client to receive, decode, and display a frame, and it takes an average of 4 ms [62].

For simplicity, we work with average delays and focus on the network delay budget of each job, D_j, considering the other delay components (which are less correlated to network management and more dependent on the particular game) as constant.

Furthermore, we assume that queuing delays at switches are negligible since our *premium* service could prioritize packets, avoiding unnecessary delays. The network delay budget is therefore spent over the links that connect the user to the game server, the resulting delay being the sum of average per-link delays d_z.

Since the time scale of our scheduling problem and the duration of the time slots is in the order of minutes, we also neglect the migration time because it is possible to obtain seamless game migration across several edge servers at millisecond timescale (cf. [62]).

b) **Bandwidth:** Focusing instead on the bandwidth requirements, we assume that each job requires a constant downlink bandwidth t_j, chosen at random from a uniform distribution with realistic bounds, while the uplink bandwidth is assumed to be negligible [422]. This assumption of constant t_j follows from the fact that it is possible to play games in streaming mode with several screen resolutions.

Furthermore, since we focus on the edge environment, we assume that the downlink bandwidth of far-edge nodes and M1 nodes is the bottleneck, rather than the per-link bandwidth, and consequently, we ignore the latter.

c) **Memory:** Each job j requires a per-time-slot memory s_j at the node where it is running. We assume that this memory requirement is known and constant for the whole duration of the job, although it randomly varies for each job.

d) **Computation requirements and energy consumption:** Both energy consumption and computation requirements of a game session are strongly correlated. In particular, we consider that there exists a linear relation between both parameters and that they may be different for each job. At a given time, we denote the energy consumption of job j as e_j, and its required computing power (in terms of processing cycles) as p_j.

We consider practical values for these requirements, extracted from some studies on gaming energy consumption [18] (in particular, from the resources in [423], [424]), such that we define both a minimum and a maximum value for both computation and energy consumption levels, as well as a mean value. Furthermore, we consider that the energy and computation for each job have a random value within the range of practical levels.

We consider two different scenarios regarding the jobs' requirements. First, we will consider that these values remain constant during the whole duration of the job. The second scenario is a practical generalization where the energy and computation requirements of a job vary over time. In such a case, we assume that the required values evolve as a random walk process constrained within the maximum and minimum values.

In general, by considering fixed computing workloads and the use of resources for each job, we make a tractable simplification that makes sense to evaluate a system in which resources are always guaranteed to the user, hence they are allocated based on the peak demand of the online game session, which makes sense for a premium service like the one studied in this chapter. It has been shown in the literature that co-locating several games at the same server that has to share un-isolable resources (e.g., GPU) leads to a general performance degradation of the QoS [22]. However, in our work, we do not consider such degradation since we do not have un-isolable resources.

With the above, we next formulate a utility optimization problem on how to allot jobs to nodes

Table 3.1: Notation used in Chapter 3

Notation	Meaning
C_j	Total Cost of job j
$C_j^{(b)}$	Energy cost (per slot) for job j
$C_j^{(d)}$	Deployment cost for job j
$C_j^{(m)}, C_j^{(p)}$	Migration and interruption costs for job j
$\mathcal{J}, J$	Set of jobs and its size
$\mathcal{N}, N$	Set of nodes (game servers), and its size
R_j	Revenue of job j
$\mathcal{T}$	Set of consecutive time slots
$\mathcal{Z}, Z$	Set of links and its size
T_n	Bandwidth of node n
t_j	Downlink throughput for job j
d_z	Delay incurred on link z
D_j	Maximum delay for job j
G_n, E_n	Green and total power at node n
e_j	Power required by job j
P_n	Computing power at node n
p_j	Computing power for job j
S_n	Memory capacity at node n
s_j	Memory required for job j
$w_{jz} = \{0,1\}$	(Variable) 1 if job j passes through link z
$x_{jn} = \{0,1\}$	(Variable) 1 if node n handles job j

to maximize the overall utility by serving as many jobs in full and minimizing total costs. This means that the use of green energy has to be prioritized, migrations should be used only if they bring more revenue than cost, and job interruptions should be avoided.

3.3. Instantaneous Utility Optimization

First, we consider the instantaneous version of our problem, meaning that revenues and costs are allocated at each time slot, every job is allocated and executed in a single time slot, and there are neither migrations nor job interruptions.

3.3.1. Problem formulation

We consider the following variables: R_j is the revenue of accepted job j while C_j is its total cost. C_j includes deployment $C_j^{(d)}$ and brown energy costs $C_j^{(b)}$ associated to the computation required for the job.

Our decision variables, denoted by x_{jn} for all $j \in \mathcal{J}$ and all $n \in \mathcal{N}$, are binary variables that indicate whether job j is allocated at edge node n ($x_{jn} = 1$) or not ($x_{jn} = 0$). w_{jz} is another binary variable, whose value is 1 if job j passes through link z and 0 otherwise.

Table 4.1 summarizes the notation used in the Chapter. The problem is therefore formulated as follows:

$$\max \sum_{n\in\mathcal{N}} \sum_{j\in\mathcal{J}} \left(R_j - C_j\right) x_{jn}; \tag{3.1a}$$

$$\text{s.t.:} \sum_{n\in\mathcal{N}} x_{jn} \leq 1, \qquad \forall j \in \mathcal{J}; \tag{3.1b}$$

$$\sum_{j\in\mathcal{J}} t_j x_{jn} \leq T_n, \qquad \forall n \in \mathcal{N}; \tag{3.1c}$$

$$\sum_{j\in\mathcal{J}} p_j x_{jn} \leq P_n, \qquad \forall n \in \mathcal{N}; \tag{3.1d}$$

$$\sum_{j\in\mathcal{J}} e_j x_{jn} \leq E_n, \qquad \forall n \in \mathcal{N}; \tag{3.1e}$$

$$\sum_{j\in\mathcal{J}} s_j x_{jn} \leq S_n, \qquad \forall n \in \mathcal{N}; \tag{3.1f}$$

$$\sum_{z\in\mathcal{Z}} d_z w_{jz} \leq D_j, \qquad \forall j \in \mathcal{J}; \tag{3.1g}$$

where:

- The objective function (3.1a) expresses the net utility;
- Constraint (3.1b) states that job j can only be allocated to a single node n;
- Constraints (3.1c) to (3.1f) ensure that a job's placement does not violate the server's capacity in terms of: downlink bandwidth (T_n), processing power (P_n), available energy (instantaneous power E_n), and memory (S_n);
- Constraint (3.1g) ensures that the average delay is guaranteed for each job;
- All weights t_j, p_j, e_j, s_j, and d_z, capacities T_n, P_n, E_n, S_n, and delay budgets D_j take positive values.

The above described problem is non-trivial to solve if no server can accommodate all jobs. In that case, the problem is NP-Hard, as shown next.

Theorem 1. *Constraints* (3.1b) *and* (3.1c) *alone make the problem NP-hard (in the strong sense).*

Proof. We reduce the Multiple Knapsack Problem (MKP) to our problem formalization. According to [425], the MKP could be written as follows: considering a set of K knapsacks with capacity W_k each, $k \in \{1, \ldots, K\}$, and a set of I items to store ($K \leq I$) where each item i has positive reward r_i and positive weight w_i, $i \in \{1, \ldots, I\}$. The objective expression is $\sum_{k=1}^{K} \sum_{i=1}^{I} r_i x_{ik}$, which has to be maximized under the constraints that $\sum_{i=1}^{I} w_i x_{ik} \leq W_k, \forall k$, and $\sum_{k=1}^{K} x_{ik} \leq 1, \forall i$, with x_{ik} being a binary variable indicating whether item i is allocated to knapsack k.

We consider the special case where $C_j = 0$ and p_j, e_j, s_j, and d_z are all equal to 1, whereas P_n, E_n, and S_n are equal to J and $D_j = Z$. In this special case, constraints (3.1d)-(3.1g) are all redundant and always satisfied.

With this special configuration, our problem is a MKP with $K = N$ knapsacks of capacity T_n and $I = J$ items with weights t_j and rewards R_j. This means that the MKP is a particular case of our problem. Therefore, we could argue that our problem is complex as much as the MKP, which is NP-hard. Since this reduction can be built in polynomial time, it follows that our problem is NP-hard. However, we highlight that due to this reduction to MKP, our problem is NP-hard in the strong sense, meaning that no polynomial-time approximation scheme is known [425] unless $P = NP$. □

3.3.2. Sub-modularity

We now show that the problem in Section 3.3.1 is sub-modular, which leads to useful performance guarantees. First, let us re-formulate the problem as a set-optimization problem. Let $\mathcal{S} \subseteq \mathcal{J} \times \mathcal{N}$ denote the set of selected single-service placements, where $(j, n) \in \mathcal{S}$ means that job j is placed at node n. Let $\Theta(\mathcal{S})$ denote the objective value of (3.1a), so that (1) becomes

$$\max \Theta(\mathcal{S}) \tag{3.2a}$$

$$\text{s.t.: } \mathcal{S} \subseteq \mathcal{J} \times \mathcal{N} \tag{3.2b}$$

$$(1b) \text{ to } (1g). \tag{3.2c}$$

Theorem 2. *The optimal value of* Θ *is a monotone increasing and sub-modular set function.*

Proof. Consider that a real-valued set function f is monotone increasing if $\forall\ \mathcal{S}_1 \subseteq \mathcal{S}_2 \subseteq \mathcal{S}$, $f(\mathcal{S}_1) \leq f(\mathcal{S}_2)$. Moreover, the function $f(\cdot)$ is sub-modular if $\forall\ \mathcal{S}_1 \subseteq \mathcal{S}_2 \subseteq \mathcal{S}$ and $u \in \mathcal{S} \setminus \mathcal{S}_2$, it holds that $f(\{u\} \cup \mathcal{S}_1) - f(\mathcal{S}_1) \geq f(\{u\} \cup \mathcal{S}_2) - f(\mathcal{S}_2)$.

The monotonicity of the solution of our problem is clear because expanding $\mathcal{S}$ (i.e., putting more jobs and/or nodes) enlarges the solution space of (3.2a) and therefore increases its optimal value. The solution is also sub-modular since, for a given amount of green energy, any increase in the number of allocated jobs will increase the amount of required polluting energy at the nodes, and therefore the overall utility obtained by including more jobs will be progressively reduced. For this class of problems, it is known that we can construct a greedy algorithm that iteratively selects the element that maximizes (subject to the constraints) the objective function, such that this algorithm achieves a performance guarantee of $1 - 1/e$ [426]. □

We present in Algorithm 3.1 the legacy `GREEDY` algorithm that solves problem (3.2) in polynomial time with performance guarantees using its submodularity property. Since the structure of the algorithm is well known and derives from [426], we omit a detailed explanation about it and refer to [426] for further information.

Algorithm 3.1 `GREEDY Algorithm`

Input: Network topology, $\mathcal{N}$, jobs $\mathcal{J}$ (with parameters $t_j, p_j, s_j, e_j, D_j \ \forall j \in \mathcal{J}$), $T_n, P_n, S_n, G_n, E_n \ \forall n \in \mathcal{N}$, $d_z \ \forall z \in \mathcal{Z}$
Output: Job-to-node placement map $\mathcal{S}$
Initialize: $\mathcal{S} = \emptyset$; $\mathcal{J}^{\emptyset} = J$; $\mathcal{S}^{\emptyset} = \mathcal{J} \times \mathcal{N}$
while $\exists (j,n) \in \mathcal{S}^{\emptyset}$ s.t. $\mathcal{S} \cup (j,n)$ satisfies (3.2c) **do**
  $(j^{\star}, n^{\star}) \leftarrow \arg\max_{(j,n)\in\mathcal{S}^{\emptyset}} \Theta(\mathcal{S} \cup (j,n))$
  $\mathcal{S} \leftarrow \mathcal{S} \cup (j^{\star}, n^{\star})$
  $\mathcal{J}^{\emptyset} \leftarrow \mathcal{J}^{\emptyset} \setminus j^{\star}$
  $\mathcal{S}^{\emptyset} = \mathcal{J}^{\emptyset} \times \mathcal{N}$
end while

3.4. Online Problem

3.4.1. Online Problem with Migrations and Penalties

The problem in Section 3.3 can be extended to the case where jobs last more than the duration of a time slot and arrive asynchronously. This situation is important because it represents the practical problem to be solved online in a real system. For this case, the objective function of the optimization problem becomes

$$\max \sum_{\tau\in\mathcal{T}} \sum_{n\in\mathcal{N}} \sum_{j\in\mathcal{J}} \Big(R_j(\tau) - C_j(\tau)\Big)\, x_{jn}(\tau) \tag{3.3}$$

where $\mathcal{T}$ is the time interval (a set of consecutive slots) over which we optimize the utility of the system, and where we recall that the total cost $C_j(\tau)$ is obtained as $C_j(\tau) = C_j^{(d)}(\tau) + C_j^{(b)}(\tau) + C_j^{(m)}(\tau) + C_j^{(p)}(\tau)$. In (3.3), we consider a one-time revenue rather than a per-time-slot revenue, such that $R_j(\tau)$ is a non-zero value only at the time slot of the arrival of request of acceptance for job j, and it does not depend on the job duration. Similarly, the deployment cost ($C_j^{(d)}$) is only non-zero at acceptance time, whereas the penalties for migration ($C_j^{(m)}$) and for job interruption ($C_j^{(p)}$) are only applied in the time slots in which the corresponding events occur.

The only term that appears in every time slot (due to its possible fluctuation) is the variable cost incurred by consuming non-renewable energy ($C_j^{(b)}$). In this optimization problem, the objective function in (3.3) must satisfy the same constraints as the problem in (3.1a), i.e., (3.1b)–(3.1g), except for the fact that these constraints have to hold at any time slot $\tau \in \mathcal{T}$.

In this new formulation, jobs can arrive in different time slots, and decisions must be made online at the slot boundaries. It is worth noting that, if migration and job interruption costs are neglected, the problem is equivalent to the one shown in Section 3.3, because it is enough to maximize the objective function slot by slot. Therefore the sub-modularity property will hold also for the online job allocation problem under such simplifying assumptions. Instead, if we consider those penalties, sub-modularity is not guaranteed. However, with realistically small migration costs and rare job interruptions, the problem can be considered, *in practice*, still sub-modular or as a small perturbation of a sub-modular case. From that, it is intuitive to consider that we can extend the greedy heuristic approach also to the online version of the problem, as shown in the following.

3.4.2. Proposed online heuristic

For sub-modular problems, it is known that the simple strategy of maximizing the instantaneous utility at each time that a new job arrives achieves high performance. This approach precludes the possibility of rejecting a job just because it might prevent the acceptance of future jobs. However, this strategy does not limit us to only using the `GREEDY` algorithm in Algorithm 3.1.

For the sake of readability, we have split the description of the `GREENING` algorithm in two sequential stages and one auxiliary function: The general description of our proposed algorithm is shown in Algorithm 3.2, which includes the entire procedure; however, the latest part of the algorithm, which handles the acceptance of newly arrived jobs, is disclosed in Algorithm 3.2-a due to space and pages limitations. Finally, a migration function called by both Algorithm 3.2 and Algorithm 3.2-a will be presented in Algorithm 3.2-m. The migration function is called in two circumstances: when an arriving job is not allocated with a direct placement and when there is a change of green energy levels due to changes in energy generation or in gaming workloads. Next, we detail the algorithm and each one of its parts.

The algorithm is triggered at the beginning of each time slot. It has two main stages. One is dedicated to react and re-schedule active jobs in the possible event that either the energy availability at the nodes or the energy requirements for the jobs change with respect to the previous time slot. The second part focuses on the admission control and optimizes the resource allocation in order to accept new jobs if it is possible.

Re-allocating ongoing jobs (Algorithm 3.2).
First, the `GREENING` algorithm checks whether the amount of available renewable energy has changed at any node. In the case in which the jobs energy and computation requirements can dynamically change, the algorithm also monitors if these values have evolved. If any of these events happen, some nodes might no longer have enough power to serve all their allocated jobs, and therefore some jobs must be migrated or interrupted.

The algorithm proceeds node by node and, for each node with not enough resources (in terms of either computation or energy resources), it examines if some job can be migrated to other—less loaded—nodes in order to avoid job interruptions. This search of both jobs to migrate and feasible destination nodes is carried out by the migration function `Migration-Greening` presented in Algorithm 3.2-m. This function takes as input a node n and a set of candidate jobs $\mathcal{J}^{(m)}$ to be migrated from node n, and it outputs which one of the candidate jobs has to be migrated ($j^{(m)}$) and toward which node is the migration conducted ($n^{(m)}$).

Importantly, before starting the search for possible migrations the nodes are sorted by the amount of *available* green energy, in descending order.[16]

For that, let us introduce some useful notations. We define $E_{n,\text{eff}}$ as the total energy required by all the jobs currently running in node n. $P_{n,\text{eff}}$ is similarly defined for the computation resources required at node n. From the definition of $E_{n,\text{eff}}$, it follows that the amount of green energy currently available at node n is obtained by subtracting $E_{n,\text{eff}}$ from the total amount of green energy

[16]Sorting nodes according to the *total* available level of green energy is also possible, as shown in our preliminary work [2], although using the residual energy is more robust to dynamic workloads.

Algorithm 3.2 `GREENING` – Proposed heuristic algorithm

Input: $\mathcal{N}$, active ($\mathcal{J}^{(\tau)}$) and new ($\mathcal{J}^{(+)}$) jobs with *current* parameters $t_j, s_j, p_j(\tau), e_j(\tau), D_j \ \forall j \in \{\mathcal{J}^{(\tau)} \cup \mathcal{J}^{(+)}\}$,
network parameters $T_n, P_n, S_n, E_n^{(\tau)} \ \forall n \in \mathcal{N}$,
$d_z \ \forall z \in \mathcal{Z}$, and previous allocation $\mathcal{S}^{(\tau-1)}$.
Output: $\mathcal{S}^{(\tau)}, \Theta^{(\tau)}$
Initialize: $\mathcal{S}^{(\tau)} \leftarrow \mathcal{S}^{(\tau-1)}$; $\Theta^{(\tau)} \leftarrow 0$;
if Jobs or Nodes energy levels change **then**
  for $n \in \mathcal{N}$ **do**
    $E_{n,\text{eff}} \leftarrow \sum_{j:(j,n)\in\mathcal{S}^{(\tau)}} e_j(\tau)$
    $P_{n,\text{eff}} \leftarrow \sum_{j:(j,n)\in\mathcal{S}^{(\tau)}} p_j(\tau)$
    while $E_n^{(\tau)} < E_{n,\text{eff}}$ or $P_n^{(\tau)} < P_{n,\text{eff}}$ **do**
      $\mathcal{J}_n \leftarrow \{\, j \mid (j,n) \in \mathcal{S}^{(\tau)}\}$
      $\mathcal{N}_g^{(\tau)} \leftarrow \texttt{sort}(\mathcal{N}, G_n^{(\tau)} - E_{n,\text{eff}})$
      $j^{(m)}, n^{(m)} \leftarrow$ `Migration-GREENING`$(\mathcal{J}_n,\ n)$
      if $n^{(m)} == -1$ {interruption} **then**
        $j^{(m)} \leftarrow \arg\min_{j:(j,n)\in\mathcal{S}^{(\tau)}}\{R_j\}$
        $\mathcal{J}^{(\tau)} \leftarrow \mathcal{J}^{(\tau)} \setminus \{j^{(m)}\}$
        $\Theta^{(\tau)} \leftarrow \Theta^{(\tau)} - C^{(p)}_{j^{(m)}}$
        $\mathcal{S}^{(\tau)} \leftarrow \mathcal{S}^{(\tau)} \setminus \left\{(j^{(m)}, n)\right\}$
      else if $n^{(m)} \neq n$ {migration} **then**
        $\Theta^{(\tau)} \leftarrow \Theta^{(\tau)} - C^{(m)}_{j^{(m)}}$
        $\mathcal{S}^{(\tau)} \leftarrow \left\{\mathcal{S}^{(\tau)} \setminus (j^{(m)}, n)\right\} \cup \left\{(j^{(m)}, n^{(m)})\right\}$
      end if
      $E_{n,\text{eff}} \leftarrow \sum_{j:(j,n)\in\mathcal{S}^{(\tau)}} e_j(\tau)$
      $P_{n,\text{eff}} \leftarrow \sum_{j:(j,n)\in\mathcal{S}^{(\tau)}} p_j(\tau)$
    end while
  end for
end if
Execute: `Acceptance-GREENING` (Algorithm 3.2-a)
$\Theta^{(\tau)} \leftarrow \Theta^{(\tau)} - \sum_{(j,n)\in\mathcal{S}^{(\tau)}} C_j^{(b)}$ {Substract energy cost}

in the node ($G_n^{(\tau)}$), where a negative value of $G_n^{(\tau)} - E_{n,\text{eff}}$ indicates the amount of polluting energy consumed at node n. Let us further denote the set of nodes ordered based on $G_n^{(\tau)} - E_{n,\text{eff}}$ as $\mathcal{N}_g^{(\tau)}$, and the node index in the i-th position of $\mathcal{N}_g^{(\tau)}$ as η_i. From this notation, it follows that $\mathcal{N}_g^{(\tau)}$ is ordered such that $(G_{\eta_i}^{(\tau)} - E_{\eta_i,\text{eff}}) \geq (G_{\eta_{i+1}}^{(\tau)} - E_{\eta_{i+1},\text{eff}})$ for any $i < N$. This ordering is motivated by the fact that nodes that have more available green energy incur less costs.

If no other node can accommodate any of the jobs in node n, Algorithm 3.2-m returns that the destination node is -1. In this latter case, when no job can be migrated, the job with the smallest revenue (since the interruption cost is comparable to the revenue) in the node is interrupted. This process is repeated until all energy and computation constraints are satisfied.

Migrating function (Algorithm 3.2-m).
The previously mentioned migration function operates a simple search on the set of potential migration destination nodes and checks the feasibility of migration based on the problem's constraints. For each candidate job j_c in the input set $\mathcal{J}^{(m)}$, we evaluate if j_c can be migrated to other node n'.

Algorithm 3.2-m `Migration-GREENING`

Input: $\mathcal{J}^{(m)}$ (Set of candidate jobs to migrate)
n (node that needs to migrate jobs)
Inherit: State and variables of Algorithm 3.2
Output: $j^{(m)}$ (job to migrate)
$n^{(m)}$ (node where $j^{(m)}$ migrates)
Initialize: $n^{(m)} \leftarrow n$
for $j_c \in \mathcal{J}^{(m)}$ **do**
  for $n' \in \mathcal{N}_g^{(\tau)} \setminus \{n\}$ **do**
    if allocating j_c to n' satisfies (3.1c)–(3.1g) **then**
      $n^{(m)} \leftarrow n'$
      $j^{(m)} \leftarrow j_c$
      break double loop over $\mathcal{J}^{(m)}$ and $\mathcal{N}_g^{(\tau)}$
    end if
  end for
end for
if $n^{(m)} == n$ {No node to migrate} **then**
  $n^{(m)} \leftarrow -1$
end if

In order to check the feasibility of the migration, the search starts from the node with more *available* green energy and the list of nodes follows by the amount of *available* green energy $\mathcal{N}_g^{(\tau)}$. In this manner, we give priority to the nodes that reduce the cost of energy consumption. The search stops as soon as a destination node is found. Once we find a node $n' \in \mathcal{N}_g^{(\tau)} \setminus n$ that can allocate a job $j_c \in \mathcal{J}^{(m)}$, we set job j_c as the migrating job ($j^{(m)}$) and node n' as the destination node ($n^{(m)}$), which are the outputs of the function. If there is no feasible pair ($j^{(m)}$, $n^{(m)}$), the function returns $n^{(m)} = -1$.

Algorithm 3.2-a `Acceptance-GREENING` – Proposed heuristic algorithm (Part II: Acceptance of new jobs)

1: Continue from line 25 in Algorithm 3.2
2: $E_{n,\text{eff}} \leftarrow \sum_{j:(j,n)\in\mathcal{S}^{(\tau)}} e_j(\tau)\ \forall n \in \mathcal{N}$
3: $\mathcal{N}_g^{(\tau)} \leftarrow \texttt{sort}(\mathcal{N}, G_n^{(\tau)} - E_{n,\text{eff}})$
4: **Define:** $\bar{\mathcal{S}}^{(\tau)}_{(j_1\to j_2),n}$ as $\{(j_2,n) \cup \{\mathcal{S}^{(\tau)} \setminus (j_1,n)\}\}$
5: **for** $j_{\text{arr}} \in \mathcal{J}^{(+)}$ **do**
6: **for** $n \in \mathcal{N}_g^{(\tau)}$ **do**
7: **if** (j_{arr}, n) satisfies (3.1c)–(3.1g) **then**
8: $\mathcal{J}^{(\tau)} \leftarrow \mathcal{J}^{(\tau)} \cup \{j_{\text{arr}}\}$
9: $\mathcal{S}^{(\tau)} \leftarrow \mathcal{S}^{(\tau)} \cup \{(j_{\text{arr}}, n)\}$
10: $\Theta^{(\tau)} \leftarrow \Theta^{(\tau)} + R_{j_{\text{arr}}} - C^{(d)}_{j_{\text{arr}}}$
11: break loop over n
12: **end if**
13: **end for**
14: **if** $j_{\text{arr}} \notin \mathcal{J}^{(\tau)}$ {New job not placed} **then**
15: **for** $n \in \mathcal{N}_g^{(\tau)}$ **do**
16: $\mathcal{J}_{\text{arr},n} \leftarrow \{\, j \mid (j,n) \in \mathcal{S}^{(\tau)}$ and $\bar{\mathcal{S}}^{(\tau)}_{(j\to j_{\text{arr}}),n}$ satisfies (3.1c)–(3.1g)$\}$
17: $j^{(m)},\ n^{(m)} \leftarrow \texttt{Migration-GREENING}(\mathcal{J}_{\text{arr},n},\ n)$
18: **if** $n^{(m)} \neq -1$ **then**
19: $\mathcal{J}^{(\tau)} \leftarrow \mathcal{J}^{(\tau)} \cup \{j_{\text{arr}}\}$
20: $\mathcal{S}^{(\tau)} \leftarrow \mathcal{S}^{(\tau)} \cup \{(j_{\text{arr}}, n)\}$
21: $\mathcal{S}^{(\tau)} \leftarrow \left\{\mathcal{S}^{(\tau)} \setminus (j^{(m)}, n)\right\} \cup \left\{(j^{(m)}, n^{(m)}),\ (j_{\text{arr}}, n)\right\}$
22: $\Theta^{(\tau)} \leftarrow \Theta^{(\tau)} + R_{j_{\text{arr}}} - C^{(d)}_{j_{\text{arr}}} - C^{(m)}_{j^{(m)}}$
23: break loop over n
24: **end if**
25: **end for**
26: **if** $j_{\text{arr}} \notin \mathcal{J}^{(\tau)}$ **then**
27: Reject job j_{arr}
28: **end if**
29: **end if**
30: **end for**
31: Continue in Algorithm 3.2

Acceptance of new jobs (Algorithm 3.2-a).
After handling the continuity of the jobs that are already in the system, `GREENING` focuses on the admission of newly arrived jobs. For that, it tries to allocate them one by one, in a sequential order. For each one of the arrived jobs, the algorithm verifies if the job fits in any of the servers. This verification follows the same *available green energy* order $\mathcal{N}_g^{(\tau)}$ as described in the previous stage, such that the nodes with the highest available green power have priority in the job allocation.

The algorithm tries a direct placement on the node at the top of the list, and moves to the next node only if the allocation is not possible according to any of the constraints. This is aligned with the greedy heuristic of the instantaneous problem, although considering just energy levels rather than overall allocation utilities. Yet, the probability of making the same decision as the greedy algorithm is high, because nodes with higher unused green energy are likely to be the ones offering the highest utility.

However, if no node in the list can take a newly arrived job, `GREENING` tries to migrate some of the already allocated jobs so that the new job can fit in the system. This section of the algorithm substantially differs from a standard greedy heuristic. In order to do this, the algorithm invokes again the migration function `Migration-Greening` from Algorithm 3.2-m on the already allocated jobs. In this case, however, there exists a difference with respect to the other call to the function. Before, the set of candidate jobs $\mathcal{J}^{(m)}$ was the whole set of jobs allocated to node n, i.e., $\mathcal{J}^{(m)} = \{ j | (j,n) \in \mathcal{S}^{(\tau)}\}$. Now, since we need to have enough space to allocate the new job, we restrict the set of candidate jobs to be composed only of the jobs enabling the new admission. This set is given by $\mathcal{J}^{(m)} = \{ j \mid (j,n) \in \mathcal{S}^{(\tau)}\} \cap \{j \mid \bar{\mathcal{S}}^{(\tau)}_{(j \to j_{\text{arr}}),n}$ satisfies (3.1c)–(3.1g)$\}$, where we have defined $\bar{\mathcal{S}}^{(\tau)}_{(j_1 \to j_2),n}$ as the resulting allocation set obtained from substituting the already allocated job j_1 by the new job j_2, i.e., $\bar{\mathcal{S}}^{(\tau)}_{(j_1 \to j_2),n} = \{(j_2,n) \cup \{\mathcal{S}^{(\tau)} \setminus (j_1,n)\}\}$.

As before, the nodes are ordered by the amount of available green energy. If the migration function does not find any migration combination that makes enough room for the new job, the job is rejected and its revenue is lost. Otherwise, the job is allocated, bringing a revenue of R_j and a cost of deployment of $C_j^{(d)}$, and the migration is committed with an incurred cost $C_j^{(m)}$.

Eventually, the algorithm discounts from the objective function the cost due to the amount of polluting (non-renewable) energy consumed during the time slot.

Note that the described migration function is greedy and so Algorithm 3.2 is still a greedy algorithm, in the sense that it makes instantaneous decisions without considering what could happen in the future. However, allowing migrations can only improve the utility obtained with a scheme without migrations, be it Algorithm 3.1 or Algorithm 3.2 simplified by skipping the call to the migration function. Therefore, we can expect that Algorithm 3.2 will offer better performance guarantees than the value $1-1/e$ of Algorithm 3.1.

To conclude, the complexity of this `GREENING` heuristic described in Algorithm 3.2 is $O(N^2J^2)$, which would reduce to $O(NJ^2)$ in case of direct placement of the arriving jobs, without migrations.

3.4.3. ETSI MEC and network slicing compatibility

In this subsection we comment on how green edge gaming is compliant to both ETSI MEC and network slicing concepts.

In the case of ETSI MEC, the MEC Orchestrator (MEO), which has an overview of the complete MEC system and therefore could be deployed in more centralized nodes, could consider the objective function (3.1a) to place games in its system. Indeed, one of the MEO's roles consists in selecting appropriate MEC host(s) for application instantiation based on constraints, such as latency, available resources, and available services [9]. To this aim, the MEO talks directly to the Virtual Infrastructure Manager (VIM), whose role is to physically deploy resources. MEC hosts provide compute, storage and network resources for the MEC applications and they could be deployed in edge and M1 nodes, where games are actually installed. Finally, games could be deployed as MEC applications, leveraging several *on-board* MEC services, such as Radio Network Information, location and traffic management to sustain the appropriate QoE level. Edge and M1 nodes could be connected through several reference points: with the MEO through a *Mm3* link and they could communicate between each other through a *Mp3* link [9]. Indeed, Figure 3.2 shows also a high level example of the edge gaming scenario implemented through ETSI MEC. Green edge gaming could also leverage network slicing to guarantee resources to servers. A service provider could reserve a slice of resource in order to satisfy end-users in terms of bandwidth computing power.

3.5. Numerical Evaluation

In this section we evaluate numerically the proposed algorithm on a set of green edge gaming scenarios, and we provide a performance comparison with alternative online gaming solutions. To perform our experiments we built a simulator with Matlab 2021a, in which we implemented our solution as well as several baselines and state-of-the-art alternatives. We study the performance of the considered solution in a set of different configurations. We are interested in analyzing how the different parameters and possible topologies of the edge computing system impact the results. For that, we run a set of experiments, where in each of the experiments we vary one aspect of the network (e.g., the arrival rate of jobs, the energy dynamics, the relation between number of far-edge nodes and M1 nodes, etc.). Among the compared algorithms, we consider cases where migrations are not considered, or where the type energy (renewable or not) is not taken into account, so as to better understand the impact of each of the features.

We start by describing the general parameters of the scenarios considered, and later we will detail each variation and its implications.

3.5.1. Simulation scenario and setup

We study the problem in a metropolitan area where users leverage online game servers in far-edge and M1 nodes, with the QoS requirements described before in terms of computing power, latency, memory, and bandwidth. For each of the settings considered in the following, we evaluate different sizes of the edge network, i.e., a set of values of the number of nodes N, always within the range compatible with the number of edge and M1 nodes that will be initially deployed in a metropolitan framework [416]. We will vary the total number of nodes between 4 and up to 48.

We simulate a green edge gaming environment during a whole day, and repeat the experiment several times until we obtain small confidence intervals. We solve Problem (3.3) with multiple

Table 3.2: Simulation parameters in Chapter 3

	Edge server	Job
Bandwidth	350 Mbps	$\mathcal{U}(10, 30)$ Mbps
Computing	3×3.5 GHz (Far-edge) 5×3.5 GHz (M1)	Random walk within 315 to 385 Mflops
Memory	3×64 GB (Far-edge) 5×64 GB (M1)	$\mathcal{U}(750, 850)$ MB
Power	1.5 kW (Far-edge) 2 kW (M1)	Random walk within 70 to 130 W
Delay	$\mathcal{U}(2, 15)$ ms	$\mathcal{U}(50, 150)$ ms
Revenue	-	$\mathcal{U}(0.03, 0.0367)$ $
Duration	-	$\mathcal{W}$eib(2504.8, 2.9637)
Deployment	0.01 $ (Far-edge) 0.015 $ (M1)	-
Migration	-	0.0003 $
Interruption	-	100% of the revenue
Energy	-	0.35 $/kWh

approaches, on a slot-by-slot basis. We consider that each time slot lasts one minute, which is much shorter than a typical online game session (~40 minutes [421]) and much longer than any job migration mechanism (lasting from tens of milliseconds [62] up to seconds) or game session launching (which takes less than a second [415]).

Network topology and server specifications

The network topology is hierarchical, as displayed in Figure 3.2, and the connectivity in between servers is assumed to be a full mesh. Throughout the experiments, we will vary the portion of the nodes that belong to the M1 type.

Server capabilities are based on a NVidia blade server [427] for edge computing. In particular, we consider that far-edge servers dedicate 3 blades to our use case, whereas M1 nodes dedicate 4 blades. Each of these blades is endowed with a CPU of 3.5 GHz for computing power, 64 GB of RAM memory and requires 450 watts (W) of energy. From this, we consider that the far-edge nodes require 1.5 kW to work at full capacity, while M1 nodes require 2 kW.

Besides, we assume that the bandwidth and incurred delays for the edge nodes are constant and in line with 5G values; specifically, we consider that each node has a downlink bandwidth of 350 Mb/s and incurs a latency of the order of 5-10 ms.

Job statistics

We assume that the time of arrival of jobs follows a Poisson process, such that the number of arrivals in each time slot is given by a Poisson random variable with rate λ. In general, we will scale the arrival rate proportionally to the number of nodes, such that λ can be generically written as $\lambda = \alpha N$, where α is a constant.

The duration of a job is extracted from a Weibull distribution, which is known to precisely characterize the distribution of the duration of online game sessions [421]. In particular, we consider a Weibull distribution with parameter $k = 2.9637$ and $\mu = 2504.8$, which yields an average duration of about 40 minutes for a typical session, and which also yields that the probability of having durations above two hours is negligible.

The computation requirement of each job ranges from 315 to 385 Mcycles/s, which amounts to 10% to 15% of a standard server CPU core. The energy requirements of the game sessions are strongly correlated with the computation requirements, and they are randomly generated within the range 70–130 W, with a mean of 100 W. These values are obtained from studies on online gaming requirements (cf. [18], [423], [424]). We provide more information about energy dynamics in the next couple of paragraphs.

In terms of bandwidth requirements, we consider that it can vary uniformly from 10 to 30 Mb/s, which matches the requirements for video resolutions that range from 720p to 4K [428]. Other game session requirements (memory, delay, CPU) are in line with previous works [14], [18]. For instance, the maximum delay allowed for each game session is a random variable uniformly distributed between 50 and 150 ms, and RAM requirements are also uniformly distributed in the interval from 750 to 850 MB.

With the above numbers, a system working *at full capacity* at all the nodes can allocate on average up to 14 jobs in each far-edge node and a maximum of about 20 jobs at each M1 node. Note that, for the already mentioned dependency on renewable energy, this peak of capacity is likely never reached in the far-edge nodes.

We would like to note that, for the assumed specifications of both nodes and jobs, the system is saturated (i.e., the servers are using all the available resources for active jobs) for $\alpha > 0.6$, while $\alpha < 0.1$ implies generically that all nodes have always room for more jobs and every job is accepted and served.

Energy fluctuation and workload dynamics

Let us explain how the energy availability at the edge nodes and the jobs' energy requirements evolve over time.

We focus first on the availability of green energy at the nodes. We consider that the green energy available at each node (and locally generated) changes every 15 minutes. In contrast with our previous work [2], we consider that the energy available presents space-time correlation. We generate random samples of green energy availability from the datasets provided by Elia for wind [419] and solar [420] energy generation. For each node, we select an energy profile from a different day of the forecasting dataset (see Figure 3.3 for a sequential visualization of the profile for seven different days).

Every 15 minutes, each node changes its available green energy following the given statistics (in terms of mean, minimum, and maximum expected value) from the random day profile. The specific value is obtained as a random sample of a PERT distribution [429] characterized by the mean, minimum, and maximum values provided by the energy profile. The PERT distribution, which is highly related to the well-known Beta distribution, is usually considered for modeling

and estimating the effect of uncertainty. The total green energy available is then the sum of both wind- and solar-generated resources.

The far-edge nodes are assumed to rely only on the local green energy available (apart from a minimum constant energy that ensures the functioning of the server). If no green energy is available at a certain time, jobs in the far-edge node must be migrated to another node or interrupted. This implies that the capacity of the far-edge nodes varies over time. Actually, in our experiments, the average capacity of the far-edge nodes ranges between 25% and 80% of the nominal capacity (i.e., between 375 W and 1.2 kW out of a nominal peak power of 1.5 kW). On the other hand, the M1 nodes always have access to the same amount of energy (2 kW), irrespective of the amount of green energy, which in our experiments is covered by green sources for up to 1.2 kW, i.e., up to 60% of the power available at an M1 node can be green.

We will consider two cases for the M1 layer: The default case (**green M1**), in which the green energy availability at M1 nodes follows the same statistics as the one for far-edge nodes. The only difference in this case between nodes is that M1 nodes use polluting energy to obtain the remaining amount of energy until 2 kW of power. Hence, they can secure a certain level of reliability in the system at the expense of a higher cost due to the cost of energy. The second case (**brown M1**) is the case in which M1 nodes make only use of the general electricity grid, i.e., they only consume non-renewable energy, while they keep enjoying the constant 2 kW. With these two cases, we try to understand the impact of heterogeneity in access to green energy resources.

Next, we describe the dynamics of the energy/power requirements for the game sessions. We also consider two different cases. The first one is the realistic scenario in which the energy requirements of a particular game session vary at each time slot (**dynamic workload**). This continuous variation is inherent to the nature of gaming. We consider that the power a job requires for the next time slot follows a random walk with standard deviation 5 W. We limit the value of this variable to the maximum and minimum values provided before (70 and 130 W, respectively), since in practice a game has a limit on both maximum and minimum requirements. The second case is the simplified case in which the energy required by a job is randomly picked at the start of the game session but then it remains constant with this initial value throughout the session (**static workload**). In both cases, the initial energy value is selected in the same way, and the dynamic scenario uses a process of zero mean, so the average energy value remains the same.This particular case is an abstraction to the approach in which the jobs are allocated the maximum amount of resources that they will ever need, such that there is no need to monitor the current demand, but at the same time implies overprovisioning and hence a waste of resources due to the inevitable variation of the real requirements, as the worst-case (maximum) value will not be frequently reached.

Monetary gains and costs

Each job provides a monetary gain R_j that ranges between 0.03$ and 0.0367$. The revenue of the job is assumed to be uniformly distributed in this range. The motivation to pay a higher fee is that higher-revenue jobs will have less chances of being interrupted. On the other hand, the migration cost of a job is fixed to 0.0003$, which is also the deployment cost at the far-edge nodes; the deployment cost for jobs allocated to M1 nodes is considered to be 50% more expensive than that of far-edge nodes, due to the longer distance. In case of a job interruption, the penalty is a

monetary cost equal to the revenue previously paid by the user (R_j). This value follows from the fact that this use case is a premium service scenario, for which the user *expects* to receive a great QoS, which would not be possible in case of interruption of an active game session.

With respect to the cost of energy consumption, we assume a price of 0.35$ per kWh, which is a realistic value in line with current prices in Europe (by first half of 2022). We consider that the energy generated from renewable sources does not incur any monetary cost, since it is locally generated at the edge node and cannot be stored for long term. Hence, only the energy from non-renewable (polluting) sources will incur the cost of 0.35 $/kWh. We assume that the energy obtained from the general grid is coming entirely from non-renewable sources.[17]

Metrics and algorithms

In our experiments, the main performance metric is the system utility, which is computed on a per time slot basis. The average system utility is proportional to the objective function of our online optimization problem (3.3), so that it represents the performance of the tested algorithms.

Besides the average utility, we also consider other metrics to shed light on the behavior of the system. For that, we also evaluate the user's QoE by means of comparing the normalized *time played*, which we define as the ratio between the sum of the service offered to all active *accepted* jobs over the total aggregate nominal duration of all (*accepted and rejected*) jobs. This metric provides us with information about the percentage of users that are satisfied with the system.

In order to provide a broader perspective of the functioning of the algorithms, we also provide the average amount of jobs in the system, as well as the amount of rejected, interrupted, and migrated jobs. We omit the study of other typical QoS metrics like jitter or packet losses because in the edge gaming scenario here considered their values are typically small and thus they are less relevant.

The above metrics are computed in terms of average and 95% confidence intervals for eight different algorithms:

- `Solver` is an algorithm that solves integer linear program (3.3), evaluated at each time slot over a time horizon of one time slot ($|\mathcal{T}| = 1$). It uses the Matlab *intlinprog* function with a timeout of 40 seconds for each simulated time slot, in order to avoid long lasting experiments. A single experiment showed in what follows can require up to one week to complete notwithstanding the imposed timeout. Due to such huge complexity, we only provide the solution with this algorithm in a subset of the experiments.
- `GREENING` is our proposed heuristic defined in Algorithm 3.2.
- `PFPJ-1` is derived from [59], which presents a resource-aware allocation and migration algorithm designed for IoT Cloud applications. It clusters servers into highly and lightly loaded subsets, and enforces migrations from highly to lightly loaded servers to enforce

[17] The portion of energy generated from renewable sources varies strongly for different countries and for different periods of the year or of the day. Our simulations are directly applicable under the assumption of mixed generation just modifying the price of the energy based on the percentage of green energy p_g as $0.35(1 - p_g)$ $/kWh.

load balancing. We have added a power constraint in the original algorithm for a fairer comparison with our scheme.

- `PFPJ-2` is the original placement algorithm defined in [59].
- `GREENING-NoMig` is a simplified version of `GREENING` where we disable the migration function, so that the heuristic becomes very similar to the baseline greedy approach of Algorithm 3.1, although with energy context information used in the sorting of candidate nodes for job allocation.
- `Random` performs probabilistic placement and does not consider migrations. It considers a random job placement with all nodes having equal probability to be chosen.
- `Free-Green` is a green-energy-aware probabilistic placement that does not consider migrations. In this case, the random job placement assigns probabilities to nodes proportionally to the level of green energy available at the node but yet not assigned to other jobs.
- `Total-Green` is a variant of `Free-Green` in which the random job placement assigns probabilities to nodes proportionally to the total level of green energy available, independently on whether the energy is already in use or not.

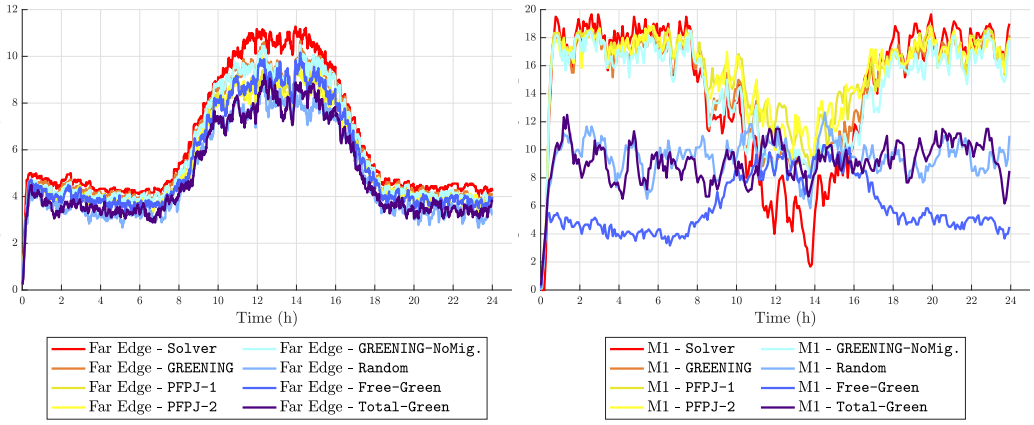
To obtain the values reported in this section, `Solver` takes several days on a Dell T640 server with 128 GB of RAM and 40 logical cores with a variable clock rate (but *intlingprog* uses only 1 thread per instance, so we parallelized the number of experiment replicas rather than the single experiment), while all other algorithms need just a few minutes.

We also made several experiments with another version of the `GREENING` algorithm, where we sort nodes according to the *total* available level of green energy (i.e. not considering if already allocated jobs are using green energy). However, in our experiments, we found out that the results were similar, even though in some cases the former `GREENING` algorithm achieved worse performance. Therefore in this Chapter, we will show only the results obtained with the `GREENING` version that considers the *available* green energy.

3.5.2. Results

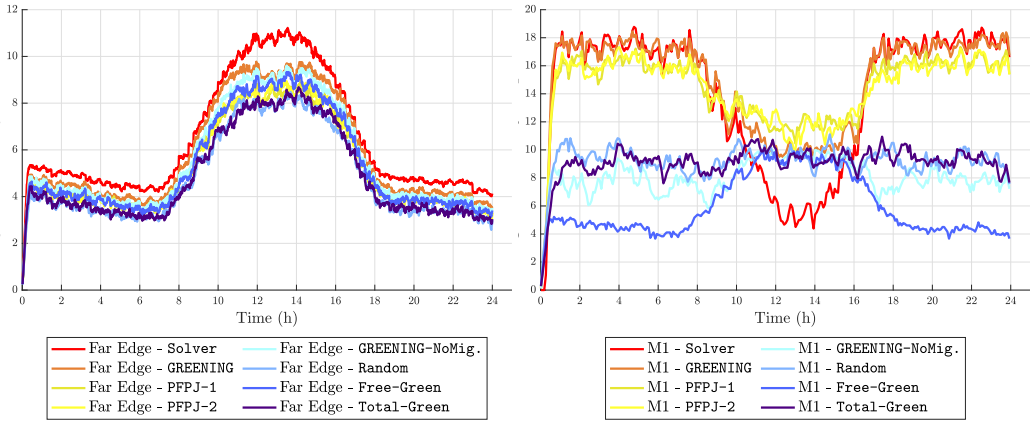
To assess and start comparing the behavior of the eight algorithms described before, Figure 3.4 and 3.5 report the average number of online gaming sessions active at an edge node over time, for a 24-hour period taken at random from the Elia's dataset. Here we use a baseline network configuration with 9 far-edge nodes, 3 M1 nodes, and a total intensity of arrivals $\lambda = 0.25N$ (expressed in terms of gaming session requests per slot). This load corresponds to a moderately high utilization of edge game resources of about 75% of the total available computing resources.

M1 nodes are allowed to use green energy according to its availability, according to the green M1 case described above. Figure 3.4a shows statistics for far-edge nodes with static workload, while Figure 3.4b refers to M1 nodes in the same experiment. The figures clearly show a dependency on the availability of green energy, which allows far-edge nodes to host more jobs in the central hours of the day. `Solver` is particularly able to offload jobs to far-edge servers as soon as possible, followed by `GREENING` and `GREENING-NoMig`. `PFPJ-1` and `PFPJ-2` perform similarly,



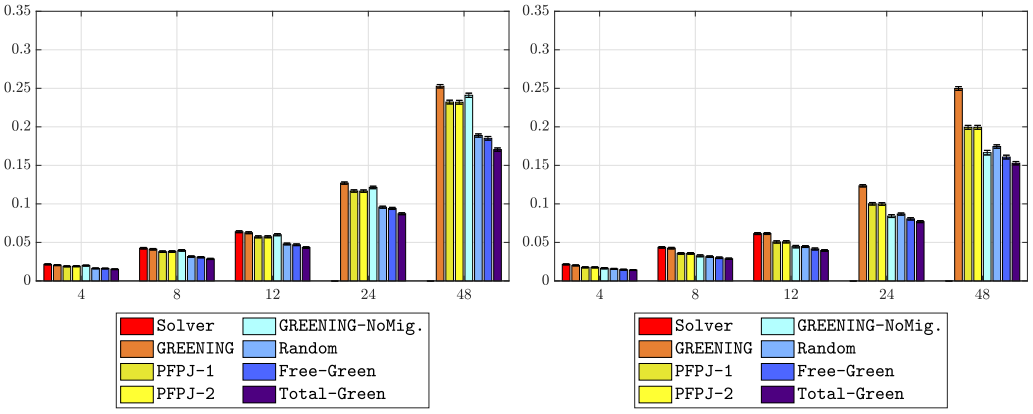
(a) Far-edge nodes with static workload (b) M1 nodes with static workload

Figure 3.4: Average number of jobs per node for an entire simulated day for the case in which the energy of far-edge nodes is 100% green whereas up to 75% of the energy available at M1 nodes can be green (but in practice only up to $\sim$ 60% in this example), with $N = 12$ nodes, 3 of which are M1 nodes, and $\lambda = 0.25N$. Static workload scenario



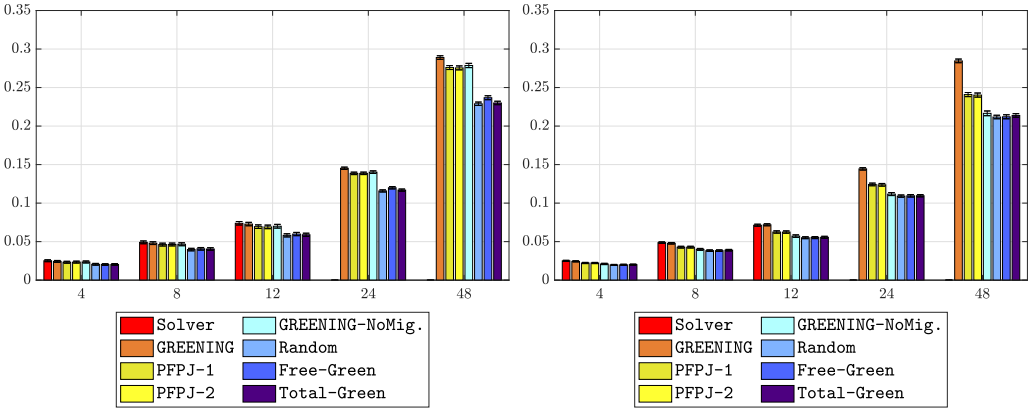
(a) Far-edge nodes with dynamic workload (b) M1 nodes with dynamic workload

Figure 3.5: Average number of jobs per node for an entire simulated day for the case in which the energy of far-edge nodes is 100% green whereas up to 75% of the energy available at M1 nodes can be green (but in practice only up to $\sim$ 60% in this example), with $N = 12$ nodes, 3 of which are M1 nodes, and $\lambda = 0.25N$. Dynamic workload scenario



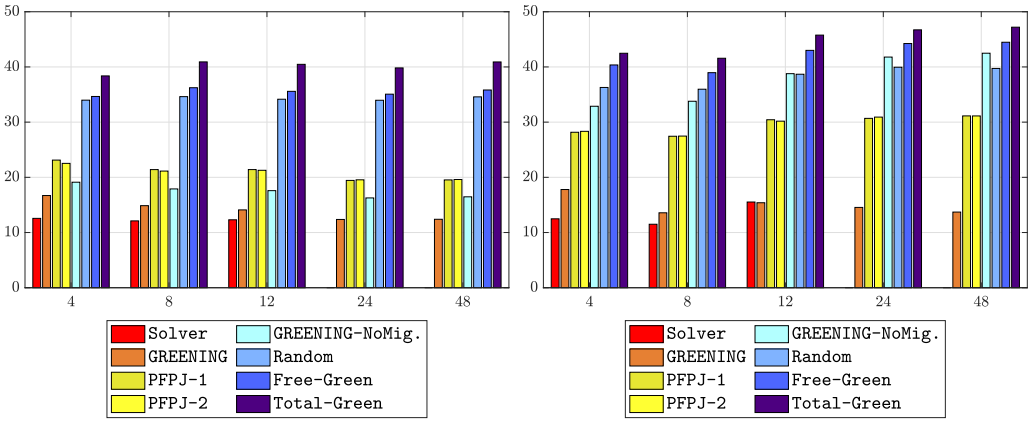
(a) Brown M1 nodes, static workload (b) Brown M1 nodes, dynamic workload

Figure 3.6: Utility comparison as the network size scales up ($\lambda = 0.25N$, far-edge to M1 node ratio equal to 3:1).



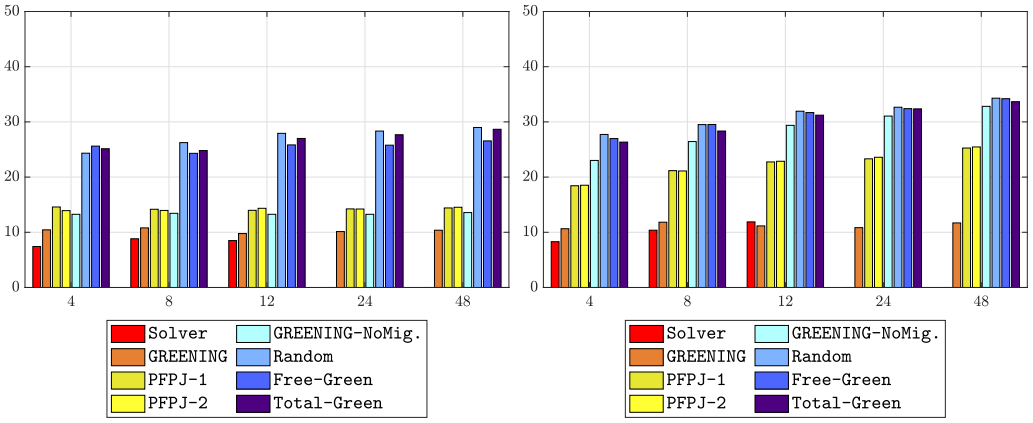
(a) Green M1 nodes, static workload (b) Green M1 nodes, dynamic workload

Figure 3.7: Utility comparison as the network size scales up ($\lambda = 0.25N$, far-edge to M1 node ratio equal to 3:1).



(a) Brown M1 nodes, static workload (b) Brown M1 nodes, dynamic workload

Figure 3.8: Utility distance from upper bound as the network size scales up ($\lambda = 0.25N$, far-edge to M1 node ratio equal to 3:1).



(a) Green M1 nodes, static workload (b) Green M1 nodes, dynamic workload

Figure 3.9: Utility distance from upper bound as the network size scales up ($\lambda = 0.25N$, far-edge to M1 node ratio equal to 3:1).

while the other algorithms are less reactive to green energy level changes. From this figure, it is clear that enforcing migrations or not has an impact, although limited. However, the way the algorithms account for the presence of green energy makes a bigger difference. Differences are further exacerbated if we consider the case of dynamic workload in Figures 3.5a and 3.5b. In this case, the average of active jobs decreases with all algorithms, which tells that gaming session deviations from the average behavior require more resources, as expected. This effect is particularly detrimental for algorithms that cannot enforce migrations. Indeed, `GREENING-NoMig` drastically reduces the number of active jobs in M1 nodes. Both under static and dynamic workloads, the `Free-Green` algorithm tends to balance the load across all available nodes, so that it is the only algorithm under which the occupancy of M1 nodes increases also in the central hours of the day. `Total-Green` behaves almost as `Random`, because the total level of green energy fluctuates for all nodes following the same daily trend. All other algorithms tend to move jobs to the far-edge when the green energy is more abundant. This also tells that the network is not saturated when the green energy level is higher although the load is quite high. Indeed, consider that, with the parameters of Table 3.2, a far-edge node can handle up to 15 jobs, on average, while an M1 node can host up to 20 jobs.

To see how the above described behaviors map onto system utility, Figures 3.6, 3.7 depicts average results for `GREENING` and the other algorithms based on the trend of Elia's traces for green energy availability as described in Section 3.5.1, for the 4 cases with green or brown M1 nodes and static or dynamic workloads. Here we fix the intensity of job arrivals per node and per slot to $0.25N$, as in Figures 3.4, 3.5 and we test different network sizes, up to 48 nodes, although for `Solver` we only report results up to 12 nodes.

All bar charts in Figures 3.6, 3.7 show that the utility increases more or less linearly with the size of the network (note that the intensity of job requests scales linearly as well), although some differences are visible. In particular, while with a tiny network scale and static workloads ($N = 4$ in Figures 3.6a and 3.7a) the differences between the 8 algorithms are small, the advantages of `GREENING` become evident as N grows and especially when considering dynamic workloads (see Figures 3.6b and 3.7b).

With static workloads, most of the algorithms perform at the same level, with `GREENING` and `Solver` only being slightly superior. This occurs because the revenue deriving from accepting a job is greater than its cost, so eventually all algorithms tend to maximize the amount of accepted jobs. They do that using direct placement, with migration only used when strictly necessary.

With dynamic workloads, the importance of timely migration becomes more evident, and performance differences emerge more clearly. For instance, the performance of `GREENING-NoMig`, which is almost as good as `GREENING` under static workloads, here decreases significantly because of the impossibility to perform migrations when the workloads are dynamic. This can cause up to a 35% of utility reduction in the case of dynamic workloads with respect to static workloads with the same average. `PFPJ-1` and `PFPJ-2` suffer dynamic workloads as well because they only enforce migrations from highly loaded to lightly loaded servers. The impairment is less evident, but it becomes substantial as the number of edge nodes increases.

Therefore, Figures 3.6, 3.7 show that well orchestrated migrations are key to achieve good results. In particular, fetching information on the dynamic properties of jobs is key to adapt the optimization on a per-slot-basis, which is what `GREENING` exploits better than the other algorithms

because it constantly adapts to changes in green energy levels and game session workloads.

Figure 3.6, 3.7 also show that the impact of having brown vs. green M1 nodes is much higher than the impact of dynamic vs. static workloads, because the use of brown energy entails high costs. Interestingly, GREENING is the only algorithm that practically performs the same with static and dynamic workloads and green M1 nodes, and its loss of performance with respect to the case with green M1 nodes and dynamic workload is very close to (or even less than) what observed for Solver. We conclude that GREENING is very robust to the network context, whereas the other heuristics suffer much more.

It it important to note that GREENING achieves practically the same utility as Solver in all cases, although with much less complexity. This means that the utility improvement achieved with Solver by offloading more jobs from M1 to far-edge nodes (see Figures 3.5, 3.4) has limited importance in most of the cases.

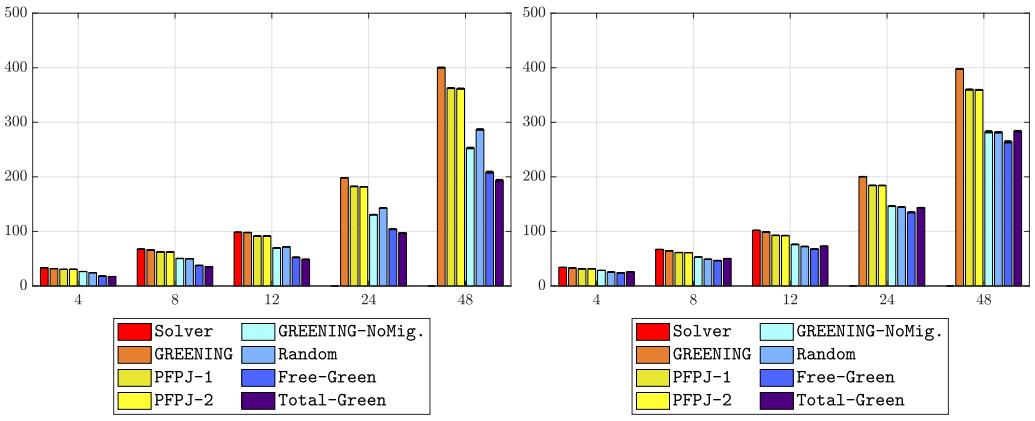
GREENING clearly performs close to Solver, which however is not guaranteed to be optimal because it solves Problem (3.3) with a very short time horizon (1 time slot only), while the results commented so far are computed for jobs lasting much longer (tens of time slots).[18] Therefore, a question that might arise at this point is: *how far is GREENING from the optimal?* To answer this question, we derive a simple upper bound on the utility. The bound is computed by multiplying the average revenue of a job, minus its deployment cost, times the average number of jobs arriving in a time slot. The above does not account for energy costs, so that we then subtract the cost of the average quantity of energy required by all jobs in a slot, but only for the part that complements the volume of energy available from renewables in the overall system. This bound is optimistic because it assumes that no brown energy is used if there is spare green energy anywhere in the system. This is definitely not the case for brown M1 nodes, and is also an overestimate on the use of green energy for any other node. In addition, the bound neglects the costs of migrations and interruptions.

Figures 3.8, 3.9 show the gap between the utility reported in Figures 3.9, 3.8 and the upper bound. The results clearly show that GREENING's distance from the (unfeasible) upper bound is slightly above 10% with brown M1 nodes and at about 10% with green M1 nodes. Solver does only slightly better, while PFPJ-1 and PFPJ-2 pay 5% to 10% more than GREENING and the other heuristics are 3 times less efficient.

As an interesting note, since GREENING and Solver are close to the lower bound, we can argue that our bound must be tight with respect to the optimal. The bound is valid not only for greedy allocations with finite time horizon, but it is valid in general over any time horizon and scheduling of jobs (including for jobs delayed before being deployed), because the bound considers the overall job arrival rate, not just the accepted jobs. Therefore we must conclude that GREENING and Solver are near-optimal with respect to any possible allocation policy, and in particular achieve much better performance than what guaranteed by using any standard greedy algorithm, which cannot guarantee anything better than $1-1/e \simeq 63\%$ of the optimal. In addition, GREENING significantly outperforms state-of-the-art algorithms in the evaluated scenarios.

The above-described results show that static workloads are preferable. However, real work-

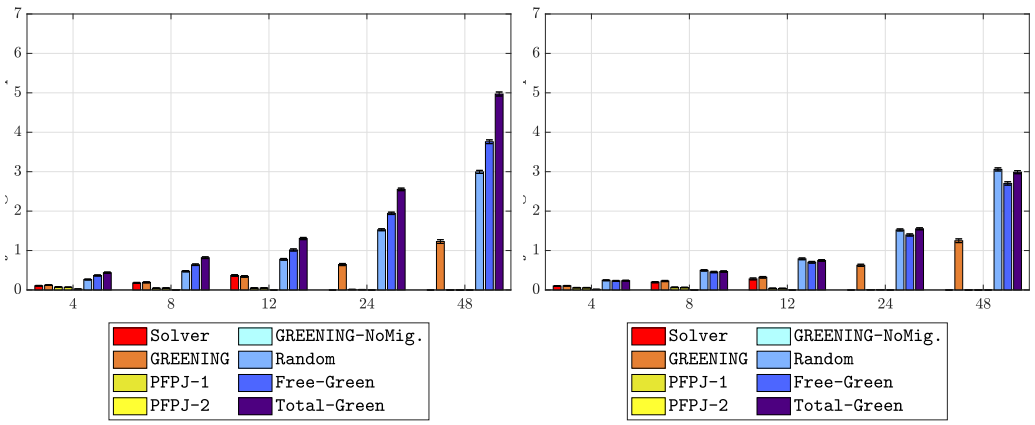
[18]Note that Solver becomes optimal as the cost of migration goes to 0 because in that case the greedy optimization of each time slot becomes optimal, as explained in Section 3.4.



(a) Brown M1 nodes

(b) Green M1 nodes

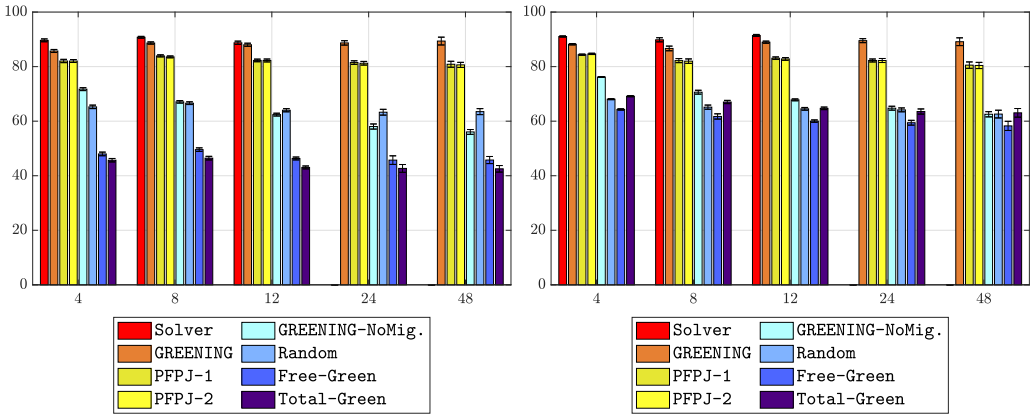
Figure 3.10: Average jobs in the system with dynamic workloads as the network size scales up ($\lambda = 0.25N$, far-edge to M1 node ratio equal to 3:1).



(a) Brown M1 nodes

(b) Green M1 nodes

Figure 3.11: Rejected jobs per time slot with dynamic workload as the network size scales up ($\lambda = 0.25N$, far-edge to M1 node ratio equal to 3:1).



(a) Brown M1 nodes

(b) Green M1 nodes

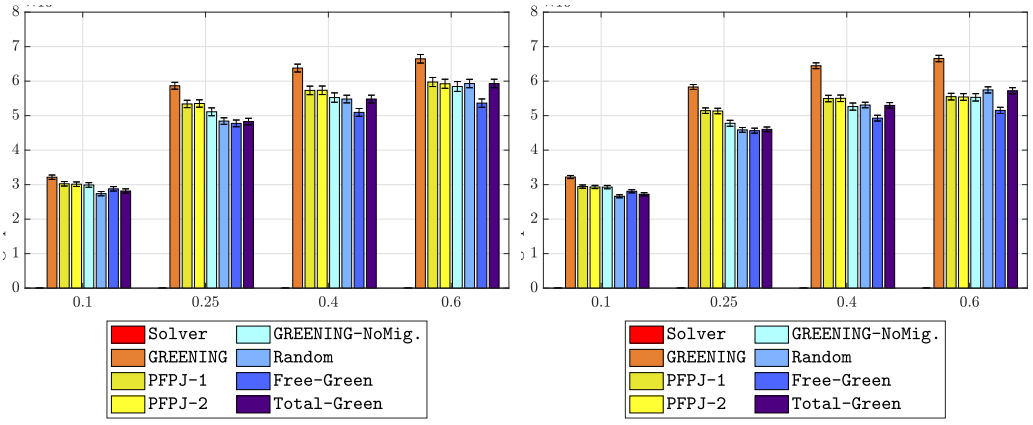
Figure 3.12: Percentage of game time played with dynamic workloads as the network size scales up ($\lambda = 0.25N$, far-edge to M1 node ratio equal to 3:1).

loads are rarely static, and can be used to idealize the service behavior under specific circumstances and with some approximation. For instance, the online game operator might want to allocate resources as the maximum possible rate required by players when there is no information about job dynamics. This is useful and provides users with strict guarantees, but incurs overprovisioning. Therefore, to further analyze the performance of `GREENING` and the other benchmarking algorithms, next we will consider results for the dynamic workload case only, which is more realistic.

The average number of jobs active in the system is depicted in Figure 3.10 for different cases and as the number of edge nodes increases. `PFPJ-1`, `PFPJ-2` and `GREENING-NoMig` perform almost as `Solver` and `GREENING`, and only `GREENING-NoMig` exhibits a significant degrade of performance. The other algorithms sustain much less jobs, on average. Therefore, the superiority of `GREENING` with respect to, e.g., `PFPJ-1`, is not due to its capacity to accept more jobs or keeping a higher number of active game sessions. It must reside instead in the ability to use resources better and avoid job interruptions.

To proceed with the analysis of the causes of utility differences between the tested algorithms, we next evaluate job rejections. Figure 3.11 shows that `Solver` and `GREENING` reject about 10% of online game session requests in all of the cases shown in the figure. For instance, with 12 nodes in total, out of which 3 are (either green or brown) M1 nodes, there are 0.3 rejected jobs per slot out of $\lambda = 0.25 \cdot 12 = 3$ requests per slot. Interestingly, `Solver` and `GREENING` can reject more jobs than `PFPJ-1`, `PFPJ-2` and `GREENING-NoMig`. The other heuristics are much worse. For instance, with brown M1 nodes, the `Total-Green` rejects about 42% of requests (e.g., with 48 nodes, the total arrival rate is 12 requests per slot, out of which about 5 are rejected), and 25% with green M1 nodes (e.g., with 48 nodes, about 3 requests out of 12 are rejected). Therefore, against intuition, high numbers of active jobs do not necessarily mean less rejected jobs. In fact, it is more convenient to reject some jobs rather than interrupt them.

Rejecting more jobs must occur because `Solver` and `GREENING` have on average more jobs allocated and less jobs interrupted, therefore having less space to admit new jobs. To evaluate the correctness of this deduction, let us observe the time played for accepted jobs, normalized to the



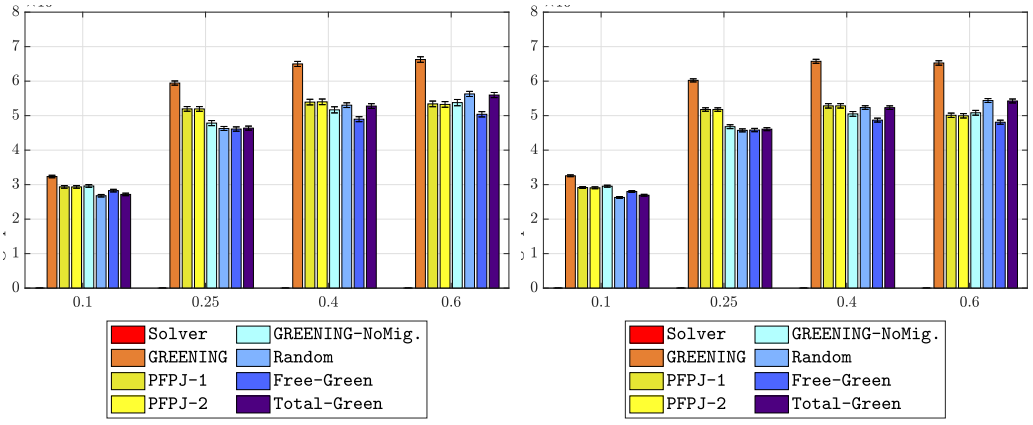
(a) $N = 4$ (b) $N = 8$

Figure 3.13: Utility comparison as the arrival rate λ size scales up for different network sizes, with dynamic workloads and far-edge to M1 node ratio equal to 3:1, and green M1 nodes.

nominal duration of all jobs, as shown in Figure 3.12. Indeed, the figure shows that `GREENING` and `Solver` guarantee the highest percentage of time played, i.e., less and shorter job interruptions. `Solver` misses about the 10% of played time, which corresponds to the 10% of rejected jobs visible in Figure 3.11. Here, `GREENING` misses an additional 2% to 3% of played time with respect to `Solver`, which is therefore due to rare job interruptions in addition to the $\sim$ 10% of rejected jobs. Other heuristics accept more jobs, but then they are forced to abruptly interrupt many more jobs, thus experiencing lower utilities. In particular, probabilistic allocation algorithms reject and interrupt many more jobs, which is because they can allocate jobs to nodes that are close to saturation. Specifically, `Total-Green` incurs about 13% to 16% of interruptions, as it yields a time played as low as 42–45% with brown M1 nodes and $\sim$ 60% with green M1 nodes (and, as noted before, about 25% and 42% of the missing played time is due to rejections for the cases with brown and green M1 nodes, respectively). Similarly, the other random algorithms cause interruptions for about 15% to 20% of the time. `PFPJ-1` and `PFPJ-2`, whose rejection rate is negligible, still have a played time of about 80%, which implies a 20% of losses due to interruptions of jobs. This hints to the fact that considering migrations only when a server is highly loaded, as done in `PFPJ-1` and `PFPJ-2` with different metrics, is more expensive and less effective than enforcing migrations continuously, as done with `GREENING` and even more massively with `Solver`.

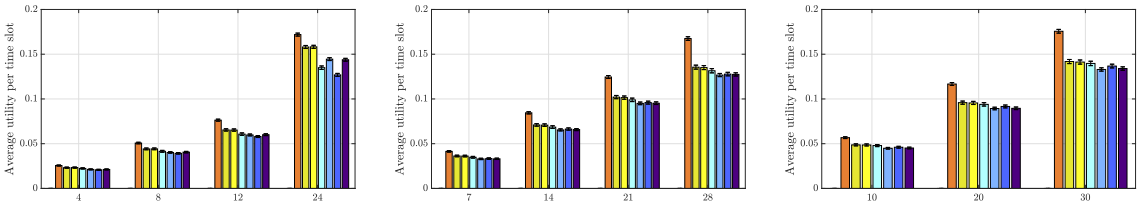
To generalize the conclusions drawn so far, we eventually evaluate the impact of the load, by varying λ and the ratio between far-edge and M1 nodes in the system.

The results of Figures 3.13 and 3.14 were obtained with 4 different levels of load and report the utility per time slot, normalized to the number of nodes. At $\lambda/N = 0.1$ requests per node per time slot, the system is underloaded and all algorithms perform similarly, with `GREENING` being a bit better than the others, for all values of N considered in the figure. The case $\lambda/N = 0.25$ is the one described before in this section, which shows that `GREENING` offers non-negligible gains, especially as the network size increases. The remaining two cases, with $\lambda/N = 0.4$ and $\lambda/N = 0.6$, are cases in which the network is lightly or heavily overloaded, respectively. In those cases,



(a) $N = 12$ (b) $N = 24$

Figure 3.14: Utility comparison as the arrival rate λ size scales up for different network sizes, with dynamic workloads and far-edge to M1 node ratio equal to 3:1, and green M1 nodes.



(a) Far-edge to M1 ratio: 3:1 (b) Far-edge to M1 ratio: 6:1 (c) Far-edge to M1 ratio: 9:1

Figure 3.15: Utility per time slot with green M1 nodes and dynamic workload at $\lambda = 0.3N$ and different far-edge to M1 node ratios.

`GREENING` is always the best choice, offering at least 20% more utility than other algorithms with at least 12 edge nodes in the system. It is interesting to note that, as the network saturates, random allocations become competitive with respect to baseline algorithms with migrations, `PFPJ-1` and `PFPJ-2`. This behavior occurs because, differently from `GREENING`, these heuristics do not attempt to minimize energy costs.

Figure 3.15 compares different far-edge to M1 node ratios, starting with the case 3:1 considered so far in this section, except here $\lambda = 0.3N$. M1 nodes are green and the game sessions exhibit dynamic workloads. The ranges for the number of nodes evaluated in each plot are different as it was not possible to use a common range that satisfied all ratios exactly. The case 3:1 is the one in which the gain of `GREENING` is the least, whereas the gain can become much higher under higher ratios like 6:1 or 9:1. Small ratios are appropriate to model early stage MEC deployment scenarios in 5G networks, due to the cost of deploying MEC hosts and controllers. Instead, much higher ratios are foreseen for future releases of 5G and beyond. Therefore, we can conclude that while `GREENING` can offer decent gain in limited deployment frameworks, its potential would be truly unleashed as the roll-out of 5G and beyond 5G networks progresses.

3.6. Summary of the Chapter

In this Chapter, we have studied the green edge gaming concept and how to maximize the utility by leveraging MEC-like facilities and locally generated green energy. We have formulated a *multi*-constrained integer-linear problem for the one-shot allocation of game sessions to servers, and shown that it is NP-hard in strong sense. The problem is however sub-modular over a single step time-horizon, and practically also over any time-horizon optimization instance, at least if migration costs are limited with respect to energy costs and/or per-job revenues. This fact might encourage the use of greedy heuristics with strict performance guarantees (of the order of $1 - 1/e$). However, we have shown that it is possible to sensibly improve performance by exploiting energy context information and timely migrations. In particular, we have shown that the green energy component is key to drive the optimization of job allocations and migrations. Moreover, a dynamic optimization is needed to account for energy level dynamics in green energy generation as well as in job power absorption. As network size and load increase, and far-edge nodes become largely prevalent in number, our proposed algorithms, `GREENING`, largely outperforms state-of-the-art approaches and achieves near-optimal results with very low complexity. Notably, without the possibility to timely migrate online game sessions, which is a feature of the edge context, the greening of online gaming could not be a viable solution.

4. GREEN AR OFFLOADING

In the previous Chapter, we presented a heuristic solution to tackle the problem of allocation and migration of jobs in a green edge gaming scenario. However, thanks to novel technological advancements in the AI field, ML algorithms (deep learning and reinforcement learning among the others) are becoming more and more sophisticated, with many research efforts devoted to applying those algorithms to a resource allocation and migration problem [63]–[65]. As we know, the development of novel use cases in beyond-5G and 6G networks will rely, among other aspects, on the availability of computing resources at the edge, therefore enabling the realization of applications that are both computationally demanding and latency-constrained, such as mobile augmented reality. In this context, applying ML algorithms becomes interesting since they *(i)* are capable of learning and adapting to the complexity of dynamic edge computing environments and *(ii)* they can adjust to changing conditions and evolving user demands, leading to more efficient resource allocation and task migration decisions. Therefore, it seems natural to try to apply ML algorithms in a scenario where tasks need to be allocated and migrated dynamically in an edge computing environment with intermittent renewable energy sources. In this Chapter, we analyze the edge operator's resource allocation to support the energy-aware offloading of MAR tasks at the edge of the cellular network with the goal of not only maximizing service acceptance (i.e., revenue) but also optimizing the operator's business utility, which depends on its carbon footprint and the profit of operating the service. We leverage Deep Reinforcement Learning to propose an efficient solution (called `GreenRL`) to operate the edge resource allocation that can adapt to different utilities. We compare our solution against baselines and another heuristic, showing how adaptability plays a key role in increasing performance.

The rest of the Chapter is organized as follows. We propose the problem of greening offloaded MAR tasks in Section 4.1 and afterward in 4.2 we show our system model. In Section 4.3 we formulate our optimization problem and we propose two different objective functions, according to the end goals of network operators. In Section 4.4 we present `GreenRL`, our DRL-based solution and finally Section 4.5 provides a numerical evaluation of our proposal while in Section 4.6 presents our concluding remarks.

4.1. Background

In beyond-5G networks, the MEC paradigm [9] places computing nodes at the edge of the cellular network, enabling new disruptive low-latency use cases. Among those use cases, we highlight Extended Reality (XR) applications [430], which cover under their umbrella both Mobile Augmented Reality (MAR) and VR applications. While the network support for the latter will be challenging even for 6G networks [13], MAR applications are becoming widespread among end-users thanks to the development of mobile equipment: a recent Huawei report [17] indicates that by 2026 the MAR market will generate over $30B in revenue, led by social apps and AR games. However, this will only be possible with the help of edge servers to offload at least partially the computation of MAR tasks [430], since many of these devices will be battery-constrained. Deploying networks

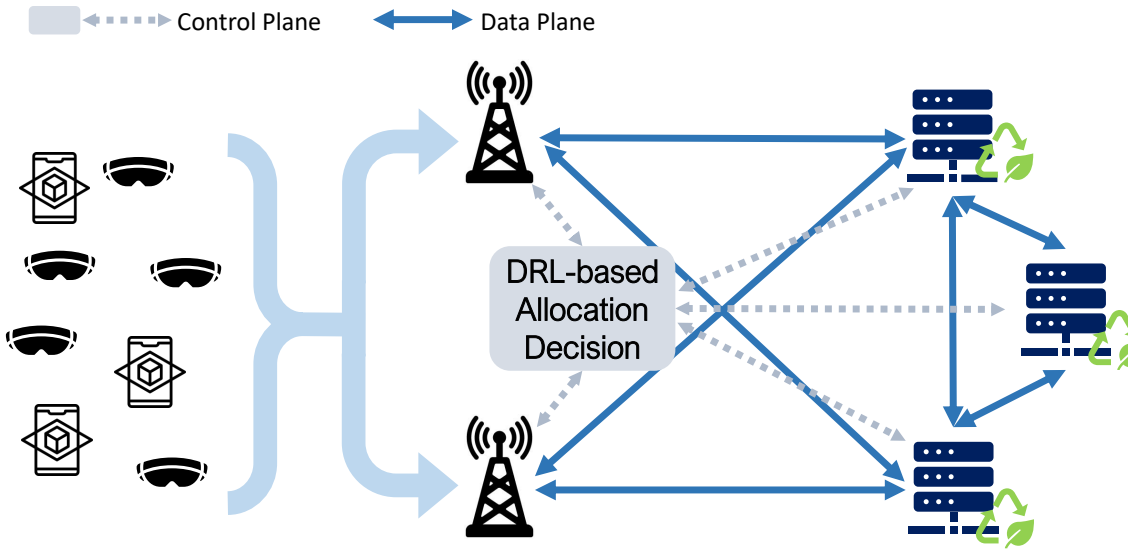


Figure 4.1: Edge scenario where MAR devices offload their computation to an edge network with migration capabilities.

that are technically capable of supporting such demanding applications (e.g., with edge computing resources) is an important challenge, but there is an even more crucial aspect to eventually see these systems deployed in real networks: sustaining the required deployment has to be profitable for operators. One manner to provide income to network operators to compensate for the large CAPEX required to deploy a MEC system is the business model based on leasing edge resources to service providers or to users on a pay-per-use model [431].

In this work, we try to answer one of the main questions arising on the topic of how to realize XR applications in next-generation communication networks: "How to distribute computation and data between different components in future XR systems?" [432], which is crucial due to the limited available resources at the edge both in computing and energy terms [1].

In particular, we study how to allocate and migrate MAR tasks in an edge network, where edge nodes have a variable amount of renewable energy. Our objective is not only to maximize the operator's profit but also to find a compromise between profit and carbon footprint, with the ultimate goal of making an edge network sustainable in both costs and energy consumption. We provide a DRL–based algorithm to propose a smart model for the allocation and migration of MAR tasks. The main contributions of this chapter are:

- This is the first work considering both the tasks' migration and the awareness of variable renewable energy at the edge.
- We optimize both profit and a weighted fair utility to compromise between profit and sustainability.
- We propose a heuristic and a DRL algorithm, which can dynamically allocate and migrate jobs according to the presence of renewable energy. We evaluate the algorithms through simulations with different loads, costs, etc., which shows that our DRL model outperforms the benchmarks and is able to adapt to different utility expressions.

Novelty and main contributions: Most of the state-of-the-art works focus on the energy

efficiency of the end-user side, while almost none of them focus on the service/operator side and, more importantly, they do not consider the impact of the availability of *intermittent renewable* energy. Indeed, this aspect can play a significant role for the operator, as nowadays non-renewable energy sources may incur exorbitant prices with high variability. In this Chapter, we fill this gap by analyzing how MAR jobs can be offloaded in an edge system dependent on both renewable and non-renewable energy and in particular focusing on how this could be sustainable for an infrastructure provider in monetary terms.

4.2. System Model

A MAR application is composed of a video source, a tracker of the user's environment position, a model for object recognition in the environment, and a rendering tool that shows the augmented world on the user's display. Except for the video source, the other tasks can be offloaded to the edge network with different latency deadlines [30]. We consider the offloading of jobs associated to the processing of video frames, as done in other works [30], [32].

Our objective is to find an allocation policy that allows the edge operator to maximize its long-term utility, where *allocation* refers to both the initial job assignment and its re-allocation (migration to another node) that might be enforced during task execution.

The utility depends on two main components: (*i*) the monetary profit, which has to be maximized, and (*ii*) the environmental footprint, which has to be minimized. These two objectives can clearly be contrasting. Thus, besides a mere cost-revenue function, we design a function that describes the inherent trade-off between profit and environmental footprint. The function is inspired on proportional fairness [433] for the normalized versions of the two unaligned objectives identified above, as we will explain in detail.

4.2.1. Network

We consider a MEC system as illustrated in Fig. 4.1, where a VNF is responsible for (re-)allocating resources every time slot of duration T_{TS}. Each edge node $n \in N$ is characterized by its maximum power consumption ($P_n^{(\max)}$) and its CPU capability ($\tilde{C}_n$), i.e., the maximum amount of processing cycles per time slot. Each MEC node is powered by renewable energy sources, which can be located on-site at the same edge node [3]. Consequently, each node has access to a variable amount of green energy that varies through time, and we assume that the amount of renewable power available at the nodes follows a generic distribution Ξ. The available green energy is variable, and the remaining power required to reach the maximum $P_n^{(\max)}$ is provided by the standard electric grid. The grid's energy source is considered to be non-renewable since a power grid fueled by a mix of renewable and non-renewable sources would only modify the relative goodness of grid energy versus the locally acquired green energy.

Offloading MAR tasks to the edge servers prevents the user from draining its battery, and the use of renewable energy sources at the edge implies that offloading tasks is an environmentally beneficial decision. Therefore, we consider that MAR tasks are by default offloaded to the edge network, provided that doing so does not entail a loss of QoE for the user, e.g., by increasing the

delay beyond a maximum acceptable threshold. If the user does not enjoy the required wireless channel conditions, its computation is done locally at the end-user device, and such the user does not request any offloading. Hence, and because we are interested in optimizing the use and allocation of computing and energy resources independently of how offloading requests are generated, we omit the modeling of the wireless network access. The impact of wireless access quality and congestion might be evaluated in future work. Finally, we consider that the edge servers are not located far away from each other, such that the fiber link connecting each other only introduces a few milliseconds of delay [434], which is below the MAR latency budget.

4.2.2. MAR tasks

We consider that each job corresponds to a MAR session requested by a user. We assume that each job has a duration ℓ_j measured in time slots of T_{TS} seconds, and $t_j^\star$ denotes the arrival time slot of job j. Each job has a required processing load that remains constant throughout the session, and the number of processing cycles per time slot required to compute job j (i.e., its size) is given by c_j. The job requests' arrival times follow a generic distribution Λ, and the size of the jobs follows a generic distribution Φ. If a job has been accepted, then it must be served without interruptions for the duration of the session, as it is assumed that this type of applications demands a great quality of service, and interruptions will not be tolerated by users paying a premium service.

4.2.3. Economic model

We consider that job j provides a revenue η_j if accepted, which is lost if it is interrupted or rejected. The revenue is assumed to be proportional to the duration and the requirements of the job. Thus, for a given fixed service fee $\bar{\eta}$ representing revenue per time slot per chunk of processing resources, job j's revenue is $\eta_j = \bar{\eta}\ell_j c_j$. The constant $\bar{\eta}$ already includes all the non-variable costs associated with the operation of the service. In this way, the only remaining OPEX to be taken into account is the variable cost of energy consumption. We consider that the locally generated green energy incurs no OPEX, but its availability is not guaranteed, as it fluctuates over time; conversely, the remaining energy obtained from the general power grid is acquired at a cost δ per energy unit and is always available. The power grid can contain a variable amount of green energy, and we model such an aspect by varying the cost δ, although we consider it to be fixed for the duration of each experiment because the price of energy from national grids changes at most every hour.

4.2.4. Decision variables

We denote the placement variable of job j at node n and time t as $x_{jn}^{(t)} \in \{0, 1\}$, such that $x_{jn}^{(t)} = 1$ indicates that job j is being managed by node n at time slot t. The processing cycles dedicated at time t for job j are similarly denoted by $c_j^{(t)}$. We further denote by J the total amount of jobs arriving in the system. Next, we formally present our metrics of interest before introducing the optimization problem.

4.2.5. Revenue metric

Let us first define $a_j \in \{0, 1\}$ as the parameter that indicates whether job j has been accepted, i.e.,

$$a_j \triangleq 1 - \prod_{n=1}^{N} \prod_{\tau=t_j^\star}^{t_j^\star+\ell_j} (1 - x_{jn}^{(\tau)}), \tag{4.1}$$

where $a_j = 1$ if job j is accepted and $a_j = 0$ otherwise, $t_j^\star$ is the arrival time of the job and ℓ_j its duration.

Since job j provides a revenue η_j, the total revenue obtained by the operator is $R \triangleq \sum_{j=1}^{J} \eta_j a_j$, while the maximum possible revenue, achieved only if all jobs are accepted, is $R_{\max} \triangleq \sum_{j=1}^{J} \eta_j$. From this notation, we define the *normalized* revenue $\bar{R} \in [0, 1]$ as

$$\bar{R} \triangleq \frac{R}{R_{\max}}. \tag{4.2}$$

4.2.6. Power consumption metric and associated cost

We assume that the power consumption derived from the computation of a job is naturally proportional to the dedicated computation resources. Specifically, job j consumes an amount of power in node n equal to $\alpha c_j^{(t)} + \gamma$ if it is served at time t, where α and γ are constant factors that translate computation capabilities to power consumption.

Furthermore, we assume that serving a job incurs an extra power cost due to the need of reconfiguration, allocation, and initialization of the resources that handle the said job. This cost appears when a job is accepted but also when a job is migrated, since, from the perspective of the node that receives the job, a migrated job is equivalent to *accept* such job in terms of resource reconfiguration. Hence, the power consumed at node n and time t to serve job j is

$$p_{jn}^{(t)} \triangleq (\alpha c_j^{(t)} + \gamma) x_{jn}^{(t)} + \beta y_{jn}^{(t)} \tag{4.3}$$

where $y_{jn}^{(t)} \in \{0, 1\}$ is 1 only if job j arrives to node n in the current time slot, i.e., it is given by

$$y_{jn}^{(t)} = (x_{jn}^{(t)} - x_{jn}^{(t-1)}) x_{jn}^{(t)}. \tag{4.4}$$

The cost of migration accounts for the resources' instantiation and management, and β is a constant factor translating such instantiation procedure into power consumption.

Since we are only interested in the consumption of *non-renewable* (-source) energy, we define the non-renewable energy consumption at node n as $p_n^{(t)}$, which is given by

$$p_n^{(t)} \triangleq \max\Big(\sum_{j=1}^{J} p_{jn}^{(t)} - g_n^{(t)}, 0 \Big) \tag{4.5}$$

where $g_n^{(t)}$ is the green energy available at node n at time t. Thus, the total non-renewable energy consumption is $P \triangleq \sum_{t=1}^{T} \sum_{n=1}^{N} p_n^{(t)}$, and the associated monetary cost is δP.

4.2.7. Migration of jobs

We consider that jobs can be migrated from one edge server to another. Specifically, we consider that, once a job (i.e., a MAR session) is allocated to one server, such server computes and sends

the video frames back to the user, e.g., either 30 or 60 frames per second (fps) at least for the whole duration of one time slot (in the order of seconds, which fits the standard time scale for network function reconfiguration [435]). At the beginning of the next time slot, based on the current network state, the VNF in charge of allocating the jobs may decide to migrate them. Since a job consists of computing video frames, there is no need for heavy data transmission between the servers: it suffices with providing the user metadata. While the new server is initiating the processes to handle the job, the initial server continues to serve the user. Once the second server is ready, the migration is effectively applied, which provides a seamless experience for the user.

4.3. Optimization problems

We consider two different utilities: pure economic profit and a proportional fairness-inspired evaluation compromising between profit and consumption of non-renewable energy. We remark that our objective is not finding the optimal allocation for a specific realization of the problem, but finding the *allocation policy* that allows the operator to maximize its long-term utility. This is important because the operator is not aware of the future job arrivals nor the future energy availability, and because of that it has to follow a policy that is based on the expected utility from current decisions. Next, we present the two corresponding optimization problems.

4.3.1. Profit maximization

The first problem aims at maximizing the operator's profit. For the sake of readability, we introduce the notations $[X] = \{1, \dots, X\}$, for any positive integer X, and $\mathcal{X} \triangleq \{x_{jn}^{(t)}\}_{j\in[J],n\in[N],t\in[T]}$. We aim at finding the optimal job allocation (and migration), i.e.,

$$\max_{\mathcal{X}} \; E_{\Lambda,\Phi,\Xi}[R - \delta P] \tag{P1}$$

$$\begin{aligned}
s.t. \quad & x_{jn}^{(t)} \in \{0,1\} && \forall\, n, j, t \in [N],\,[J],\,[T] && (4.6)\\
& \sum_{n=1}^{N} x_{jn}^{(t)} = a_j && \forall\, j \in [J], t \in [t_j^* : t_j^* + \ell_j] && (4.7)\\
& \sum_{j=1}^{J} c_j^{(t)} x_{jn}^{(t)} \le \tilde{C}_n && \forall\, n, t \in [N],\,[T] && (4.8)
\end{aligned}$$

where (4.7) states that, if a job is accepted, it can only be allocated to one node at each time slot, and (4.8) is the node computation constraint, which ensures that the sum of processing resources allocated at a node n is at most equal to its processing capacity.

We also define the profit margin $\bar{B}$ as the ratio between the profit $R - \delta P$ and the total *potential* revenue $R_{\max}$, such that $\bar{B} \triangleq \frac{R-\delta P}{R_{\max}}$.

4.3.2. Joint optimization of revenue and carbon footprint

To be able to jointly optimize such disparate metrics as power consumption and revenue, we define a normalized version of the two metrics, such that both are enclosed in the range between 0 and 1.

First, for revenue, we consider its normalized expression defined in (4.2), given by $\bar{R} \triangleq \frac{R}{R_{\max}}$, $\bar{R} \in [0, 1]$. For the power metric, we define the *normalized power saving*, which takes the form:

$$\bar{P}_{\rho_p} \triangleq 1 - \rho_p \frac{P}{P_{\max}}, \tag{4.9}$$

where $P_{\max}$ is defined as the maximum possible non-renewable power consumption, i.e., as $P_{\max} \triangleq \sum_{t=1}^{T} \sum_{n=1}^{N} (P_n^{(\max)} - g_n^{(t)})$, and where $\rho_p \in [0, 1)$ is a weight to prevent degenerate cases (since $\bar{P}_{\rho_p} \geq 1 - \rho_b > 0$) and to balance the importance of power in the objective function. $\bar{P}_{\rho_p}$ can be seen as the percentage of non-renewable energy that we can *save* to the worst-case scenario. Note that $\bar{P}_{\rho_p} \in (0, 1]$ is maximized when the non-renewable power consumption P is minimized.

Furthermore, we introduce a weight $\rho_r \in [0, 1)$ for the revenue term, which aims at tuning the contribution of the revenue to the objective function, such that the final revenue metric is

$$\bar{R}_{\rho_r} = \rho_r \bar{R} + (1 - \rho_r), \tag{4.10}$$

which is a linear mapping from $[0, 1]$ onto $(1 - \rho_r, 1]$. The closer the value of ρ_r to 1, the bigger the range of the metric is and thus the more importance it has for the operator.

The two coefficients ρ_r, ρ_p allow us to masquerade or emphasize the contribution of each of the metrics and to avoid that they take value 0, which would cause instability problems due to the logarithmic shape of the function defined next. Their values will depend on the relative importance that each of the two metrics has for the operator. We make use of the proportional-fair rule to jointly optimize both metrics because it ensures that the best solution is such that the sum of relative improvements for each term achieved by any other solution is below zero [433], which leads to maximizing $\log(\bar{P}_{\rho_p} \cdot \bar{R}_{\rho_r})$. Hence, our optimization problem is

$$\max_{\mathcal{X}} \; E_{\Lambda,\Phi,\Xi}[\log(\bar{P}_{\rho_p} \cdot \bar{R}_{\rho_r})] \tag{P2}$$

$$s.t. \;\; (4.6),\; (4.7),\; (4.8) \tag{4.11}$$

4.3.3. Complexity Analysis

We analyze the complexity of both (P1) and (P2), proving that they are NP-hard. For that, we prove that the well-known 0-1 Knapsack Problem (KP) can be reduced to our problem, i.e., that every instance of the KP can be transformed into an instance of our problem. We first recall the definition of the KP.

Definition 1 (0-1 Knapsack Problem[425])**.** Consider a knapsack with capacity $\tilde{C}_n$ and J jobs, where job j has profit p_j and weight w_j. Let $x_j \in \{0, 1\}$ denote the variable representing whether job j is introduced in the knapsack ($x_j = 1$). Then, the KP is defined as

$$\underset{\mathcal{X}}{\text{maximize}} \; \sum_{j=1}^{J} p_j x_j \tag{4.12}$$

$$s.t. \quad x_j \in \{0, 1\} \qquad \forall j \in [J] \tag{4.13}$$

$$\sum_{j=1}^{J} w_j x_j \leq \tilde{C}_n \tag{4.14}$$

Theorem 3. *The problems* (P1) *and* (P2) *are NP-hard.*

Table 4.1: Notation used in Chapter 4

Notation	Description
a_j	$a_j = 1 \Leftrightarrow$ job j is accepted, $a_j = 0$ otherwise
$c_j^{(t)}$	Required processing for job j at time t
$\tilde{C}_n$	CPU capabilities at node n
J	Number of jobs
N	Number of nodes
$p_n^{(t)}$	Non-renewable used power at node n, time t
$g_n^{(t)}$	Available green power at node n at time t
P	Total non-renewable power consumption
$P_n^{(\max)}$	Total available power at node n
$\bar{P}_{\rho_p}$	Normalized non-renewable power savings
R	Total revenue
$R_{\max}$	Maximum revenue (if all jobs are accepted)
$\bar{R}$	Normalized revenue $\bar{R} \triangleq \frac{R}{R_{\max}}$
$\bar{R}_{\rho_r}$	Normalized revenue metric
$t_j^\star$	Arrival time slot of job j
$y_{jn}^{(t)}$	Indicates if job j arrived to node n at time t
Λ	Distribution of job's arrival time
Φ	Distribution of job's size
Ξ	Distribution of green power availability
α, γ, β	Constant power–computation factors
$x_{jn}^{(t)}$	Placement variable of j at node n at time t

Proof. Since (P1) and (P2) only differ in the objective function, we can simultaneously prove both. We start by considering a specific case of our problem, which takes the following assumptions:

1. We consider a single time instant ($T = 1$).
2. We consider a single node ($N = 1$).
3. We consider that $c_j \leq \tilde{C}_1$ for all j.
4. We consider that $g_n^{(t)} \geq P_n^{(\max)}$, for any t.
5. Λ, Φ, Ξ are deterministic and known constants.
6. All jobs last a single time slot ($\ell_j = 1$).

We can remove the expectation over Λ, Φ, Ξ in the objective functions because from 5 we know the number and size of all jobs. From 1-2, we can omit the sub-index n and the super-index (t).

Furthermore, 4 implies that $p_n^{(t)} = 0$, and thus $P = 0$ and $\bar{P}_{\rho_p} = 1$. Hence, the objective function of (P1) becomes $\max_{\mathcal{X}} R$ and that of (P2) becomes $\max_{\mathcal{X}} \log(\bar{R}_{\rho_r})$. Due to the monotonicity of the log function, the values that maximize $\log(\bar{R}_{\rho_r})$ are the same ones that maximize $\bar{R}$; since we are interested in the argument that maximizes the function rather than the maximum value itself, we can consider $\max_{\mathcal{X}} \bar{R}$ as our objective function for (P2) in this specific case. Since $R_{\max}$ does not depend on the decision variables, we can substitute the objective function in (P2) by $\max_{\mathcal{X}} R$, which matches that of (P1). Hence, in this particular setting given by 1–6, both (P1) and (P2) are equivalent. Since (4.1) and the assumption $N = T = 1$ imply that $R = \sum_{j=1}^{J} \eta_j x_j$, our problem is equivalent to

$$\max_{\mathcal{X}} \ \sum_{j=1}^{J} \eta_j x_j \tag{4.15}$$

$$s.t. \quad x_j \in \{0, 1\} \qquad \forall j \in [J] \tag{4.16}$$

$$\sum_{j=1}^{J} c_j x_j \leq \tilde{C}_n \tag{4.17}$$

By assigning $p_j \leftarrow \eta_j$, $w_j \leftarrow c_j$ in (4.12), the KP can be reduced to this specific case of our problem. Hence, we can argue that (P1) and (P2) are as complex as KP, which is NP-hard. Since this reduction can be built in polynomial time, both problems are NP-hard. □

4.4. Algorithms

The previous optimization problem is the formal definition of the objective of the operator. In real network deployments, the operator cannot know how many jobs are going to arrive in the incoming time slots. Because of that, it has to rely on probabilistic policies, which determine the best decision to take at the current moment based on the expected behavior of the system. Furthermore, even if the operator had access to future samples, the NP-hardness of the optimization problem would discourage any attempt to directly solve it, as the complexity and required time for iteratively solving such a problem would not be acceptable in a real-time system.

Thus, we need to derive practical algorithms to provide a solution for the problem above. As previously stated approaches based on RL are usually considered for these decision-making

problems, where we cannot obtain an optimal solution and the objective depends on the previous and future decisions. For the application of RL, the problem is typically modeled as a MDP, and, consequently, we reformulate the problem as an MDP.

4.4.1. Reformulation of the Problem as a Markov Decision Process

The scenario presented in Section 4.3 can be stated as a MDP through the 4-tuple (S, A, P_a, R_a) governing any MDP: The *state space* (S), *action space* (A), the probability distribution of the next state given the current state and action a ($P_a(s'|s)$), and the immediate reward from arriving to state s' from state s due to action a ($R_a(s, s')$).

Agent

The agent corresponds to the edge orchestrator controlling the N edge nodes. At every time slot, it must take J_t decisions, where J_t is the number of jobs that are present at the beginning of time slot t, including the jobs already being served and the new requests arrived since the last time slot.

State space

The state space is comprised of all the information obtained by the agent from the environment that influences the action of the agent. In a given time slot, the state S_t (also called observation) indicates the current value of each one of the variables of interest. Our state space is composed of three different parts: the load of each of the edge nodes, the green energy availability at each edge node, and, finally, an indicator stating whether the next job to be managed is a new arrival request or is already being served by one of the edge nodes. Thus, it contains $2N + 1$ dimensions.

To facilitate the learning convergence, and because RL performs better when dealing with discrete variables, we consider a quantized status of both load and green energy availability. Next, we describe the possible state values and how we discretize the variables.

For the green energy availability, we consider a three-step quantization: state 0 means that there is enough renewable energy to fit more jobs, state 1 that in the current state, all energy consumed is renewable but there is not enough to serve a new job and state 2 that the node is already taking energy from non-renewable sources.

Regarding the nodes' load, we quantize the amount of processing cycles required by the node to compute the allocated jobs in a non-linear way. Specifically, we consider that the quantization step follows a logarithmic progression, such that the steps become smaller as the node is more loaded. This follows from the intuitive idea that the exact load is not so important when the node is handling a low computation load, but it becomes more important when it is approaching maximum capacity and thus consuming more energy. The quantization is done to enforce that the last step represents the case where the node cannot accept more jobs and the previous step indicates that there is space for at least one job.

Finally, the last state dimension is a discrete variable that can take $N + 1$ values, from 0 to N, where 0 represents that the job is newly arrived (and thus it can be rejected or allocated), whereas

Algorithm 4.1 GreenRL: Admission control and resource (re)allocation at each time slot

Input: state S_t, set of jobs in the system ($J^{(m)}$), set of new arrived jobs ($J^{(+)}$), green energy $g_n^{(t)}$,
Output: Allocation decisions for time slot t
for $j \in J^{(m)}$ **do** { Migration decisions}
  Agent selects action $a = \pi(S_t)$.
  Check constraints violation ((4.8) or job's interruption (4.7)).
  If positive, agent receives a penalty, episode is stopped.
  Evaluate reward and update next state.
end for
for $j \in J_{(+)}$ **do** { Acceptance decisions}
  Agent selects action $a = \pi(S_t)$.
  Check constraint violation (4.8).
  If positive, agent receives a penalty, episode is stopped.
  Evaluate reward and update next state.
end for

a value V from 1 to N indicates that the job is already being served and it is currently placed at node V (such that it can be migrated to other node but cannot be interrupted).

Action space

The action space is the description of the agent's decision. In our scenario, the agent decides whether to accept, reject, or migrate each job, which translates in our model to an action space composed of a single discrete variable that can take the values $\{0, 1, \ldots, N\}$. A value $a \in \{1, 2, \ldots, N\}$ indicates that the job is allocated on node a for the next time slot. This value can represent either an allocation of a new job or a migration to a different node in the case of already served jobs if $a \neq S_t(2N + 1)$. Finally, $a = 0$ represents that the job is rejected; consequently, $a = 0$ is only allowed for new jobs as active sessions must not be interrupted.

4.4.2. Deep Reinforcement Learning-Based Solution: GreenRL

We present next the proposed DRL-based solution, denoted as GreenRL, and whose high-level description is presented in Algorithm 4.1.

As described above, at the beginning of each time slot t the operator decides for each job present in the network. The operator first handles the jobs that are currently being served in the system. Those jobs must be served until they finish, but they can be migrated from one node to another, which would incur a migration cost as described in Section 4.3. The agent evaluates job-by-job the possibility of migrating, and once they have been managed, it starts deciding whether the newly arrived jobs are accepted or not, and where to allocate them. If accepted, the job provides revenue to the operator (unless it is later interrupted).

We note that it is not trivial to correctly define an adequate reward function since the state de-

pends on decisions taken several time slots in advance due to the shorter time scale of the resource re-allocation time slot (few seconds) to the duration of the jobs (many minutes). Because of that, we consider that the reward is a normalized sliding-window version of the objective functions defined in Section 4.3, as is detailed in the following.

- *Profit*: To compute the reward, we first calculate the profit obtained in the last T_r time slots, computed as the difference between the revenue provided by the jobs accepted in those T_r time slots and the energy cost generated by *all* the jobs in the system during such interval. Then, we normalize this profit by the total revenue of all the jobs that arrived during the T_r time slots, i.e., the reward corresponds to the value of $\bar{B}$ for the last T_r time slots and lies in the range $[0, 1]$.
- *Fairness*: Similarly, the reward depends on the revenue and cost during the last T_r time slots. Since reward normalization is known to help to achieve better performance for DRL algorithms, instead of computing $\log(\bar{P}_{\rho_p} \cdot \bar{R}_{\rho_r})$ as indicated in (P2), with lies in the range $[\log((1-\rho_r)(1-\rho_p)), 0]$, the reward is given by $\log(\bar{P}_{\rho_p} \cdot \bar{R}_{\rho_r} + 1)/\log(2)$, such that it only takes values in the $[0, 1]$ interval.

Training is split into episodes, each one including up to a maximum number of time slots, and up until the agent takes a decision that violates any of the physical constraints (i.e., it interrupts a job that has been accepted or it allocates to a node more computing resources than its maximum capacity). When an episode is terminated due to a constraint violation, the last reward is set to -1 to prevent the agent from repeating the mistake. Once trained, the agent is modeled through a probabilistic policy $\pi(S_t)$.

In our work, we leverage an algorithm called Asynchronous Advantage Actor Critic (A2C) [436] and, in particular, we use the implementation provided by Stable Baselines3 library[19]. A2C is based on Actor-Critic policy gradient methods, and its main idea is the use of an asynchronous updating scheme that operates on fixed-length segments of experience, executing asynchronously multiple agents in parallel. We refer to the original paper for further details [436].

`GreenRL` may take actions that lead to QoE disruptions: It could (*i*) reject jobs when there was green energy available for them, (*ii*) allocate jobs to a node that is already full, (*iii*) interrupt an ongoing job. The training process must learn to avoid all these cases.

4.4.3. Heuristic Algorithm: `GreenH`

We also propose a heuristic algorithm to compare it with the performance of the DRL-based approach. The goal is to understand the benefits that DRL can bring over designed algorithms that do not suffer from the low performance that the system can face during (possibly long) training periods. This heuristic algorithm, to which we refer as `GreenH`, also handles in-system job migrations and is aware of the green energy distribution across nodes.

The algorithm acts similarly for both in-system jobs and new jobs: It computes the unused green energy at each node, i.e., the available green energy at the node minus the current node

[19] https://stable-baselines3.readthedocs.io/en/master/modules/a2c.html

consumption due to the jobs processing. Then, it allocates the job to the node with more unused green energy. When all nodes are consuming polluting energy, the decision is random. A job can be forcibly rejected or interrupted if the selected node runs out of computing resources.

4.4.4. Baselines

We also evaluate the performance of two simple baselines. These algorithms do not implement migration of in-system jobs across nodes, and they do not take into account the distribution of green energy, i.e., they focus on maximizing only the operator's revenue. Consequently, the two algorithms accept all the incoming jobs, and if the node does not have enough computing power, the job is forcibly interrupted with the subsequent loss of user QoE. These two baseline algorithms are defined as follows.

- `Random`: This algorithm selects randomly the node to which each job is sent. The decision is drawn from a uniform discrete distribution of range $\{1, 2, \ldots, N\}$.
- `Emptier`: sends the job to the node that has the lowest load among the N nodes. If there are several nodes with the lowest load, the choice is uniformly random among such nodes.

4.5. Numerical Evaluation

We evaluate numerically the four previously described algorithms on a set of network scenarios and varying parameters. Besides these four solutions, we also provide the optimal solution obtained by solving the optimization problem (P1) or (P2). We remark that this last result, to which we refer as `Solver`, is an *ideal* solution that is not feasible since to solve the optimization, we must consider that we know in advance the state of the system in the future time slots (number of arrivals, energy availability, etc.). `Solver` is also impractical due to the complexity of the problem, since its computation takes a time that is several orders of magnitude bigger than the actual operation time. We built a Python simulator, where the DRL framework is built on Stable-Baselines3 library and the optimal solution for `Solver` is built using Python's SciPy library. We performed our experiments in a Dell T640 server with 128 GB of RAM and 40 logical cores.

4.5.1. Simulation scenario

We evaluate a MEC system as the one presented in Fig. 4.1, where all edge nodes are interconnected in a full-mesh topology and have the same computing capacity and maximum power consumption. We consider that each edge node offers a computing capacity of $\tilde{C}_n = 2$ TeraFLOPS, which is in line with first MEC deployments in metropolitan areas [3]. Time slots last 5 seconds, which is thus the longest a user would wait to start a session. The value of the factors to transform computation to energy consumption are $\alpha = 0.9, \beta = 0.1$ and $\gamma = 0.1$, such that the additional cost of migrating a job is approximately a 10% of the cost of computation per time slot.

We consider MAR jobs as a sequence of video frames. The edge nodes process video frames with resolution 800×800, which is the same order of magnitude usually considered in the litera-

ture [32], [34], [45]. According to [32], this video frame size requires 20% of the total computing resources of a 2-TeraFLOPS edge server.

We assume that a job requires a constant computation per frame. Considering dynamic video frame sizes, which would vary the computation requirements, could be investigated in future work. We assume a static session length of several minutes,[20] much longer than the slot length, and that jobs' arrivals follow a Poisson process.

We assume a set of default values for all the parameters that are valid for all the experiments unless stated otherwise, and which are provided in Table 4.2. In some cases, we consider that the revenue unit $\bar{\eta}$ is smaller than the cost unit δ (of energy coming from the power grid) due to several reasons: (*i*) operators' profit margin per unit of service is known to be very small, (*ii*) the final cost of the service is smaller due to the (relatively) free use of local green energy, and (*iii*) naturally if the revenue of a job is always bigger than its cost, the decision will be simpler because all jobs will be accepted, and the only aspect that will matter is *where* to allocate them. The experiments are evaluated by averaging at least 20 different realizations with different renewable energy realizations. Each energy realization is independent of each other to obtain a comprehensive analysis covering all the possible energy distributions.

Training of DRL solution `GreenRL`

The DRL solution is built upon the well-known A2C algorithm. We evaluated also other state-or-the-art DRL approaches, such as Proximal Policy Optimization (PPO) or Deep Q-Learning (DQN), but they were underperforming for all the experiments, and hence we do not include them in the results. As training parameters, we consider a total training duration of 10 million decisions (although it is enough to train for 1 million steps for simple cases, e.g., when $N = 3$), a maximum length per episode of 1024, a learning rate of 0.0007, and a batch size of 8. Learning rate and batch size are selected after a careful evaluation, and they lie within their typical range. Both policy and value neural networks have the same architecture: each one is defined as a MLP network composed of two hidden layers of 64 neurons each.

4.5.2. Results

We provide the results of the described experiments. In the figures, all the vertical bars represent the 95% confidence interval. We also omit the transient phase from all the experiments.

Trade-off of fairness function

As previously mentioned, the operators may be interested in optimizing different Key Performance Indicators (KPIs) besides the pure economic benefit, KPIs that are more aligned with high-level goals of the company such as satisfying certain environmental objectives, e.g., the flexible utility introduced in (P2). Yet, the optimal decisions for (P2) will strongly differ depending

[20] For longer sessions, we can assume that, once the session has reached a certain duration, a new offloading request is made to renew the service.

Table 4.2: Simulation parameters in Chapter 4

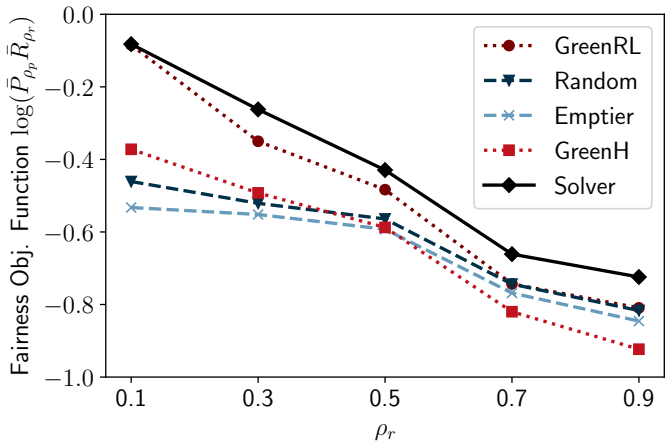
Job length (time slots)	7
Arrival Rate (Per time slot)	3
Job computation resources	20% of server capacity $\tilde{C}_n$
$\bar{\eta}$ **(Revenue/time slot/flop)**	10
δ **(Cost/time slot/flop)**	15
Renewable Energy	Random Uniform $\mu = 0.5P_n^{(\max)}$
Fairness weights	$\rho_r = 0.4, \;\; \rho_p = 0.95$

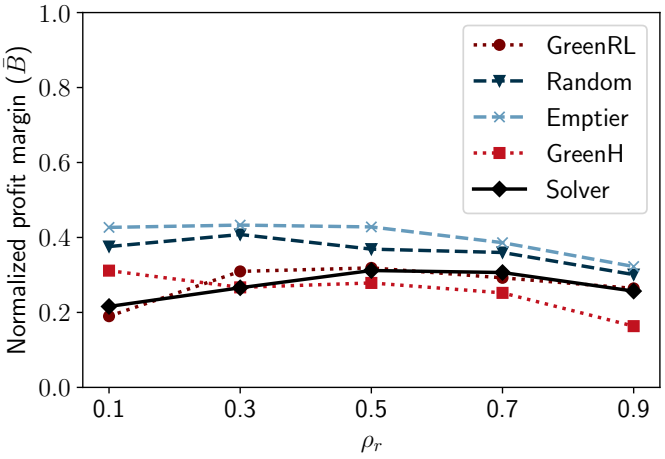
on the weights ρ_r, ρ_p that are best suited for the operator's objective. To understand the impact of these parameters, we evaluate the performance of the algorithms for different values of $\rho_r \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ for a fixed value of $\rho_p = 0.95$. We evaluate a scenario with 3 edge nodes, serving a set of users whose requests amount to an average load of 50% of the total capacity of the nodes. In this experiment, we consider that the revenue per job is 20% *higher* than the cost of computing such job without local green energy, i.e., $\bar{\eta} = 1.2\delta$. Yet, due to the varying weight of the revenue term in (P2), it is not always better to accept all the jobs.

The results are shown in Figure 4.2. Figure 4.2a represents the main objective function (that of (P2)) for which both `Solver` and `GreenRL` are optimized. We observe how `GreenRL` performs close to the (ideal) `Solver`, outperforming the other algorithms by more than 20% except for $\rho_r \geq 0.7$. For $\rho_r \geq 0.7$, `GreenRL` performs as well as the best of the other algorithms because, with the considered level of green energy, accepting all the jobs is almost optimal. Figure 4.2b represents the resulting normalized profit margin $\bar{B}$, which is the ratio between the profit $R-\delta P$ and the total *potential* revenue $R_{\max}$, such that $\bar{B} \triangleq \frac{R-\delta P}{R_{\max}}$. We recall that the algorithms are optimized to maximize the fairness-like expression (P2), and not the profit (P1). `GreenRL` and `Solver` perform worse than the baselines for this metric, but that is expected since they aim to optimize the other metric. Indeed, the relative result with respect to the baselines worsens as the weight of the profit (ρ_r) decreases.

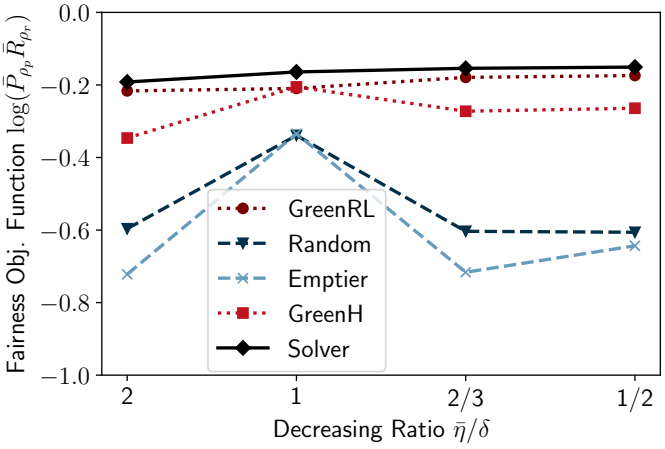
Impact of revenue/cost ratio

The cost of non-renewable energy, which may vary greatly, also impacts the performance of the edge network. Figure 4.2c shows how changing the relation between revenue and cost could affect the performance of all algorithms, for the case where `Solver` and `GreenRL` are optimized for (P2) ($\log(\bar{P}_{\rho_p}\bar{R}_{\rho_r})$). `GreenRL` is able to perform really close to the solver's performance for any cost, and with an important gap w.r.t. to the baselines. The good result of all the algorithms when $\frac{\bar{\eta}}{\delta} = 1$ is due to the fact that, with that value, accepting a job that consumes only non-renewable energy has the same profit as rejecting the job, and thus different decisions can lead to similar performances.





(b)



(c)

Figure 4.2: Evaluation of the performance of the algorithms as function of several system parameters for problem (P2). We present in (a) the value of the objective function of (P2) for different values of the weight of the revenue metric ρ_r, and in (b) the corresponding value of normalized profit margin obtained in this case (when we do not directly optimize the profit). In (c), we show the impact of varying the ratio between revenue ($\bar{\eta}$) and cost (δ) again when solving (P2).

Baseline network topology with 7 nodes

Figure 4.3 shows the results for a network topology of 7 edge nodes that abides by the parameters of Table 4.2, except for the jobs arrival rate, which is set to $\lambda = 4$, such that the average system load is 65%. We provide a detailed analysis, whose results are summarized in Fig. 4.3, by presenting four metrics of interest: (*i*) the cost of energy (Fig. 4.3a), (*ii*) the normalized profit margin $\bar{B}$ (Fig. 4.3b), (*ii*) the portion of users accepted, rejected and interrupted (Fig. 4.3c), and (*iv*) the use (and excess) of *green* and *polluting* energy (Fig. 4.3d), and for the two considered problems: The bars labeled as "Fairness" correspond to the case where the algorithms are designed to optimize (P2), whereas the cases labeled as "Profit" when we optimize for (P1). Fig. 4.3a-4.3b show how, in the "Fairness" case, `GreenRL` attains the same profit as when it is optimized to maximize the profit, while also greatly reducing the power cost, which endorses the consideration of (P2). In fact, `GreenRL` for (P2) reduces the energy cost by more than 80% w.r.t. `GreenH` and the same `GreenRL` optimized for profit, 95% w.r.t. the baselines, and matches that of `Solver`.

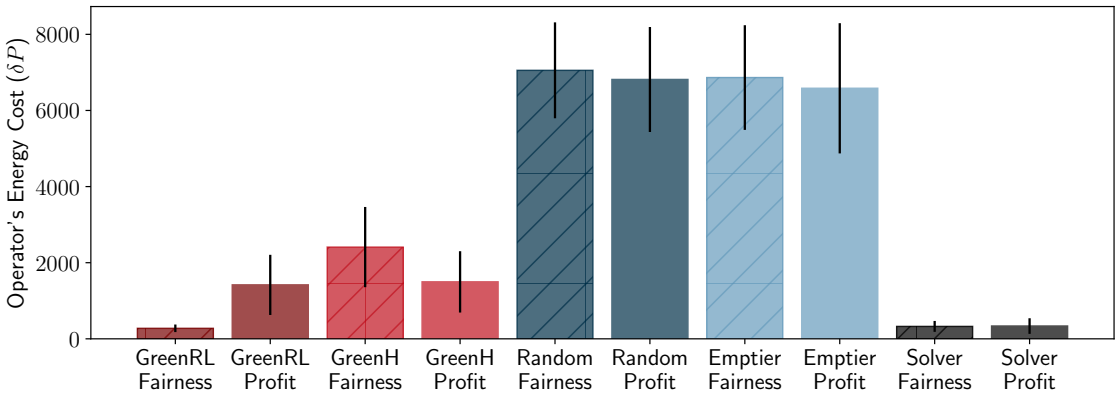
Fig. 4.3c indicates that `GreenRL` is the most conservative algorithm, since it accepts the smallest number of jobs, but it does so to ensure that no job is interrupted. We remark that, in our model, not accepting a job is not critical, as it is then computed at the user device (with the only drawback of draining its battery), whereas interrupting a job that was offloaded has a huge impact in the end-user QoE, since due to the nature of the MAR sessions it is highly probable that the user is unable to continue the session. The explanation of this conservative behavior is also complemented with Fig. 4.3d: `Random` is shown in Fig. 4.3c to be the most aggressive, and Fig. 4.3d shows that such approach is detrimental because it incurs high consumption of non-renewable energy. Similar rationale can be applied, to a lesser extent, to `Emptier`, and while both `Random` and `Emptier` accept more jobs, they also incur more jobs interruptions and power consumption. Moreover, `GreenRL` performs quite close to `Solver`, which is also quite conservative, although it is able to accept more jobs. In terms of underuse of green energy, the algorithms perform similarly, except for `Solver`; yet, `GreenRL` is the only one that does not use non-renewable energy in place of green energy. The patterns in energy sources usage are also maintained for the case with $N = 10$ edge nodes, represented in Fig. 4.4.

`GreenRL` and `GreenH` enjoy a great performance because they allow migration to nodes with more available green energy. Specifically, throughout all the experiments here presented, `GreenRL` migrates an average of 10% of the jobs in the system, while `GreenH` migrates 25% of the jobs. The amount of migrations adds a monetary cost for the network operator, thus giving an additional reasons why `GreenRL` migrate less.

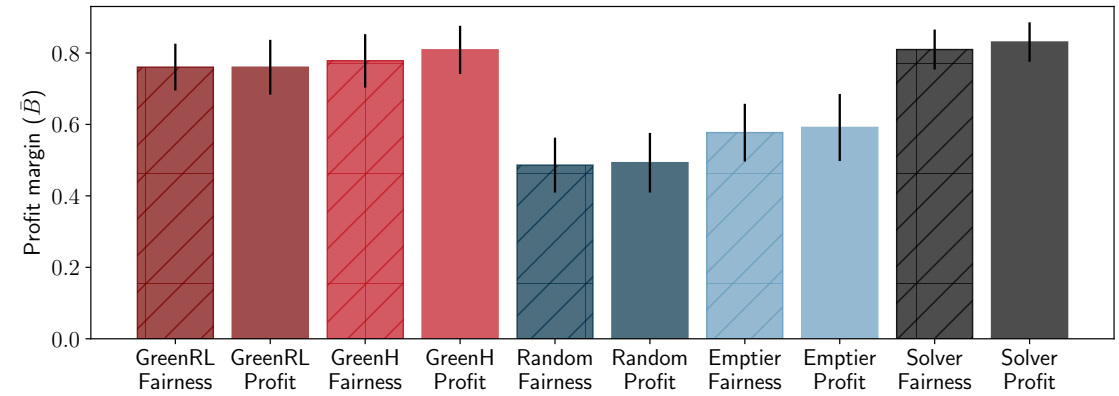
Performance as function of the number of edge nodes

We evaluate the performance obtained by the algorithms when optimizing (P1) (Fig. 4.5) and (P2) (Fig.4.6) as function of the number of edge nodes. Due to its intractable complexity, `Solver` is evaluated only up to 10 nodes. For (P1), all algorithms have similar performance except for `Random` since, to maximize profit under the considered parameters, especially the low-to-medium load, the optimal choice is almost always accepting the job.

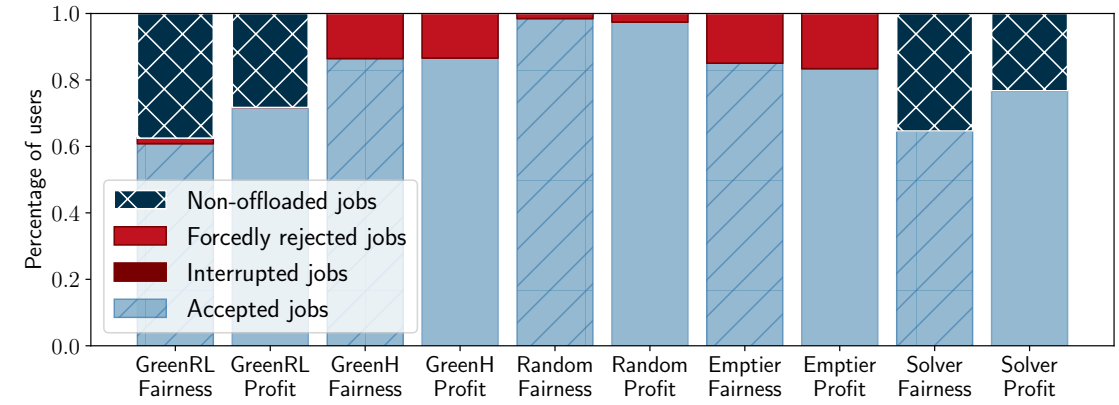
Instead, in Fig. 4.6, differences are more pronounced as `GreenRL` and `GreenH` outplay the



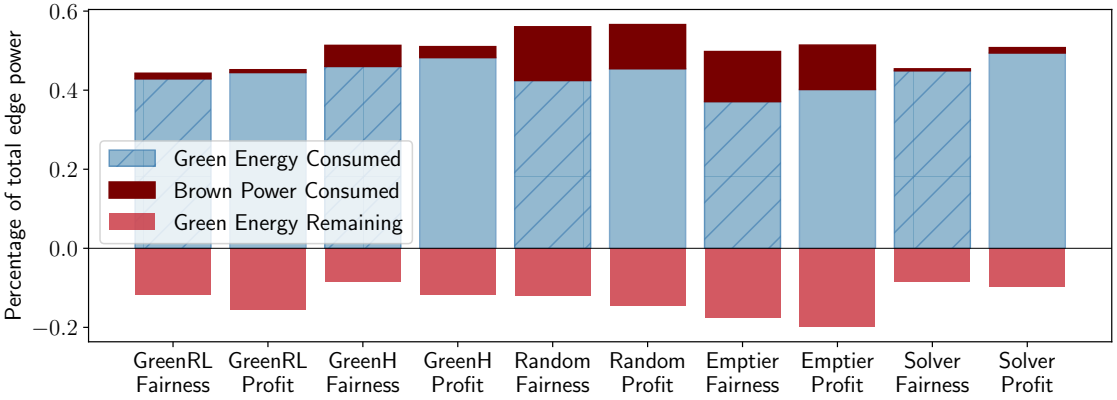
(a) Mean energy cost and standard deviation for the different algorithms



(b) Mean profit margin $\bar{B}$ and standard deviation for the different algorithms



(c) Type of service provided to the users



(d) Energy use by type of energy source and wasted (unused) green energy

Figure 4.3: Performance for the scenario with $N = 7$ nodes. We represent the results obtained when solving both the problem (P1) (labeled "Profit") and (P2) (labeled "Fairness").

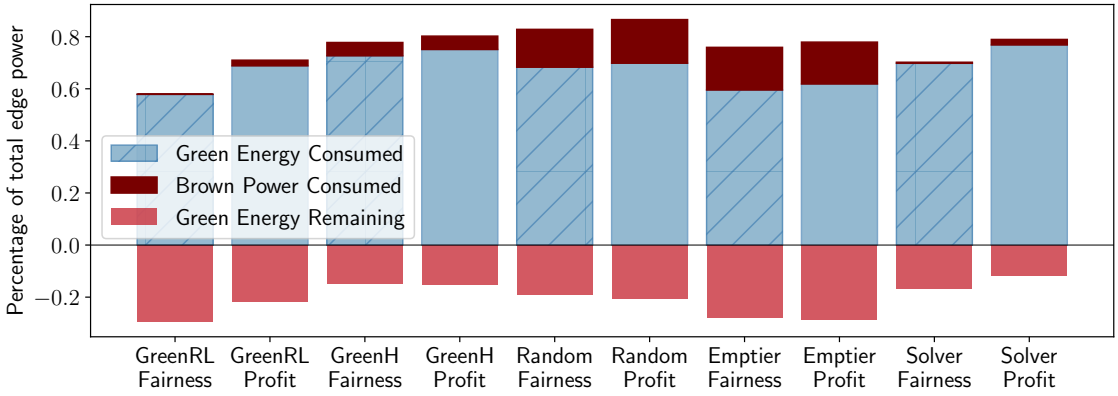


Figure 4.4: Energy use by type of energy source and wasted (unused) green energy for the scenario with $N = 10$ nodes.

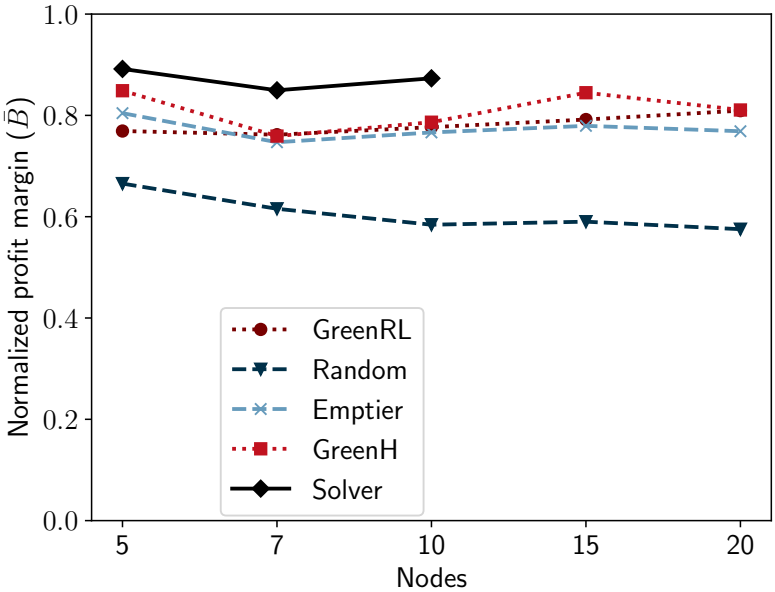


Figure 4.5: Performance as function of the number of nodes for (P1). Computing load is 65% of the total capacity.

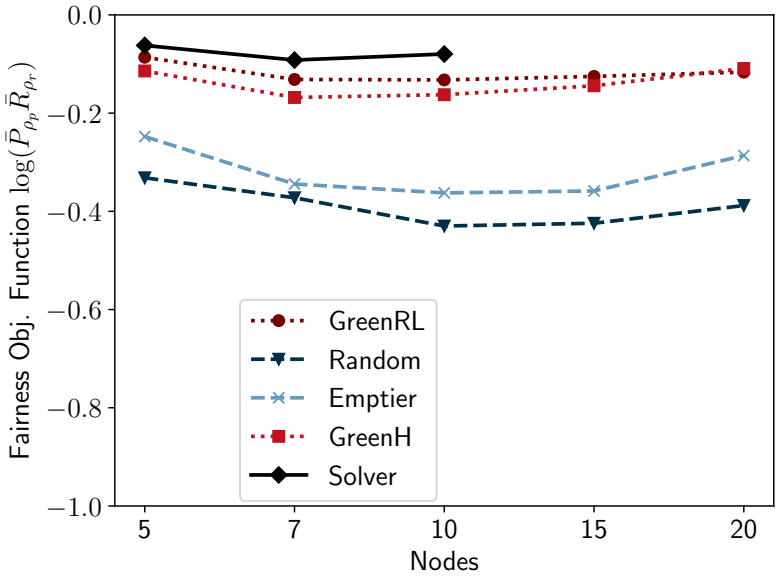


Figure 4.6: Performance as function of the number of nodes for (P2). Computing load is 65% of the total capacity.

baselines, with `GreenRL` being the best algorithm, tied with `GreenH` for $N = 20$. This is another evidence of how having a DRL-approach adapting its decisions based on different factors could lead to more robust and scalable solutions in MEC settings.

Impact of higher loads with low green energy

Finally, we also evaluate the performance of `GreenRL` in the case where the edge network processes a higher load while sustaining low availability of green energy. For all experiments, we consider the 5-edge-node scenario with an average load of 96% of the maximum capacity of the system and a green energy distribution being uniformly random between 10% and 40% of the maximum required energy. For this scenario, we considered two values of δ for a fixed revenue $\bar{\eta} = 10$.

We report the results in Table 4.3. Again, `GreenRL` greatly outperforms the other algorithms for problem (P2), achieving a performance which is within the confidence interval of the `Solver`'s one. Instead, for (P1) with cost $\delta = \bar{\eta}$, the simplest baseline performs as good as `Solver`. As previously mentioned, this is due to the fact that the operator obtains the same profit by accepting jobs that only consume non-renewable energy as it does by rejecting them. However, when costs increase, `GreenRL` stands out again.

4.6. Summary of the Chapter

In this chapter, we have analyzed the offloading of MAR tasks in an edge scenario where the edge nodes have variable availability of renewable energy sources, and we have proposed a DRL-based algorithm that can adapt the decisions to the current energy availability and energy costs, as well as to different business utilities. We have proposed a flexible utility that offers a trade-off

Table 4.3: Result with high load and low green energy level

Objective →	Fairness (P2)	Fairness (P2)	Profit (P1)	Profit (P1)
Cost (δ)→	10 ($\frac{\bar{\eta}}{\delta} = 1$)	15 ($\frac{\bar{\eta}}{\delta} = \frac{2}{3}$)	10 ($\frac{\bar{\eta}}{\delta} = 1$)	15 ($\frac{\bar{\eta}}{\delta} = \frac{2}{3}$)
GreenRL	**-0.410 ± 0.128**	**-0.396 ± 0.019**	**0.332 ± 0.032**	**0.180 ± 0.035**
GreenH	-0.803 ± 0.044	-0.689 ± 0.035	0.194 ± 0.041	0.095 ± 0.041
Random	-0.934 ± 0.096	-0.777 ± 0.041	0.313 ± 0.026	0.126 ± 0.036
Emptier	-1.285 ± 0.144	-1.012 ± 0.080	**0.341 ± 0.035**	0.103 ± 0.043
Solver	-0.378 ± 0.102	-0.331 ± 0.018	0.348 ± 0.102	0.280 ± 0.034

between pure net economic profit and the minimization of non-renewable energy consumption (and, consequently, carbon footprint). The proposed approach can adapt the admission control, resource allocation and migration depending on the state of the network, and we have proven through simulations that the model achieves performances close to an ideal optimal solution. We have also shown how job migrations between edge nodes can help to sustain the MAR business model at the edge, which motivates further analysis to understand if migrations also benefit when considering, e.g., the latency of the wireless link or a comprehensive energy model that includes the end devices and the energy consumption due to the data transport.

5. CONCLUSIONS

5.1. Summary and Conclusions

In this thesis, we have investigated novel solutions to support the allocation and migration of tasks in an edge computing (i.e., MEC) scenario, where constrained edge servers partially or completely depend on the presence of intermittent renewable energies. Indeed, the presence of edge computing will be pivotal in sustaining demanding use cases such as AR or VR in future cellular networks (e.g., 6G networks). However, providing computing resources at the edge of a cellular network infrastructure brings novel challenges for network operators, and therefore novel techniques and frameworks should be addressed by the research community.

In Chapter 2 we overviewed the state-of-the-art on the MEC deployment in an edge-cellular ecosystem. We concentrated on mainly two aspects: We first overviewed standards and in particular, the ETSI-MEC standardization. Leveraging standardization brings several benefits: for instance, using only standardized interfaces removes the complexity of connecting different interfaces from different vendors, upgrading general performance. We discussed the general ETSI MEC framework and how it will be implemented in 5G networks. Secondly, we focused on several techniques or novel scenarios that the edge computing paradigm will enable: we explored the computation offloading paradigm, one of the main use cases enabled by edge computing, migration techniques (or virtual machines/containers migration) to follow users' mobility, and how to (flexibly) deploy MEC resources at the edge. We also addressed how important verticals (Automotive, Smart City, Media, Smart Factories, and eHealthcare) are leveraging the presence of computing resources at the edge. Finally, we studied a high-level scenario on how different verticals could be supported at the same time by an edge computing infrastructure in a smart metropolitan scenario. The findings of this study enabled us to discover or give important insights for the directions of this thesis. In particular, we highlighted how difficult it is (for an edge infrastructure) to sustain many users using computing-intensive verticals such as AR/VR and how the costs of deployment and maintenance of the whole edge computing infrastructure are big, even for a small one.

Motivated by these issues, we proposed and studied intelligent algorithms that could allocate and migrate tasks within close-edge servers to maximize the revenues of edge operators (i.e., maximizing the admittance of users but also decreasing eventual costs). One way to decrease costs is to leverage renewable energies, which however are intermittent and not always present in day-to-day life. In the following two chapters of the thesis, we studied those scenarios applied to different verticals.

In particular, in Chapter 3 we studied the scenario of green edge gaming, where cloud gaming sessions are moved to edge computing infrastructure with the benefit of decreased latency but also with the drawback of intelligently allocating tasks due to the scarcity of edge computing resources. Furthermore, those computing capabilities depend on the presence of renewable sources (the more green energy there is, the more powerful are these servers), and therefore tasks should be allocated and migrated according to that volatile presence. We propose `GREENING` a smart heuristic that can move and allocate jobs according to the presence of green energy, maximizing the revenue stream

for edge operators while decreasing the use of brown (i.e., costly) energy.

Along the same lines, we studied the green offloading problem for augmented reality applications. Indeed, in Chapter 4 we studied the offloading, allocation, and migration of MAR tasks at the edge, where servers again depend on green energy. In this problem, we leveraged a proportional fairness structure for our optimization problem to find a compromise between revenues (e.g., admitting as much as tasks as possible) and carbon footprint (i.e., decreasing as much as possible the usage of brown energy) and we proposed a DRL-based solution, showing how ML approaches can help solve complex problems and can find a sweet spot in a proportional fairness structure compared to heuristics or simple baselines.

5.2. Future work

From this thesis, several lines could be considered interesting for future work:

- In our works, we considered the presence of energy only on the edge servers-side, since we were interested in targeting the performance from a network operator point of view. However, it could be interesting to study an **end-to-end** system, considering, therefore, the energy consumption of devices, base stations, and edge servers. This end-to-end approach has still not been evaluated in the literature.
- In this thesis, we considered different verticals separately. However, as also started in Chapter 2, it would be interesting to extend our problems to a **Green Multi-Vertical** optimization scenario, where several verticals at the same time need to leverage edge servers partially dependant on renewable sources. Verticals have distinct requirements, therefore needing different optimization time-scales and computing resource requirements, among other challenges. This could for instance involve developing multi-objective optimization problems to balance the needs of different verticals while maximizing overall system efficiency and revenue generation.
- It would be interesting to design experiments with our solutions, using a real-world architecture. This would help in gathering novel insights.

BIBLIOGRAPHY

[1] F. Spinelli and V. Mancuso, "Toward Enabled Industrial Verticals in 5G: A Survey on MEC-Based Approaches to Provisioning and Flexibility," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 596–630, 2021. DOI: 10.1109/COMST.2020.3037674.

[2] F. Spinelli and V. Mancuso, "A Migration Path Toward Green Edge Gaming," in *2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM) (WoWMoM 2022)*, Belfast, United Kingdom (Great Britain), Jun. 2022.

[3] F. Spinelli, A. Bazco-Nogueras, and V. Mancuso, "Edge Gaming: A Greening Perspective," *Computer Commun.*, vol. 192, pp. 89–105, 2022. DOI: https://doi.org/10.1016/j.comcom.2022.05.022.

[4] F. Spinelli, A. Bazco Nogueras, and V. Mancuso, "Offloading Augmented Reality Tasks with Smart Energy Source-Aware Algorithms at the Edge," in *Proceedings of the Int'l ACM Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWiM '23, <conf-loc>, <city>Montreal</city>, <state>Quebec</state>, <country>Canada</country>, </conf-loc>: Association for Computing Machinery, 2023, pp. 73–82. DOI: 10.1145/3616388.3617523. [Online]. Available: https://doi.org/10.1145/3616388.3617523.

[5] F. Spinelli, L. Iannone, and J. Tollet, "Multi-Cloud Chaining with Segment Routing," in *2020 IFIP Networking Conference (Networking)*, 2020, pp. 514–518.

[6] S. Yi, C. Li, and Q. Li, "A survey of Fog Computing: Concepts, Applications and Issues," in *Proceedings of the 2015 Workshop on Mobile Big Data*, ser. Mobidata '15, Hangzhou, China: ACM, 2015, pp. 37–42. DOI: 10.1145/2757384.2757397. [Online]. Available: http://doi.acm.org/10.1145/2757384.2757397.

[7] M. Satyanarayanan, V. Bahl, R. Caceres, and N. Davies, "The Case for VM-based Cloudlets in Mobile Computing," *IEEE Pervasive Computing*, Nov. 2009. [Online]. Available: https://www.microsoft.com/en-us/research/publication/the-case-for-vm-based-cloudlets-in-mobile-computing/.

[8] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, and B. Weihl, "Globally distributed content delivery," *IEEE Internet Computing*, vol. 6, no. 5, pp. 50–58, 2002. DOI: 10.1109/MIC.2002.1036038.

[9] ETSI, "Multi-Access Edge Computing (MEC); Framework and Reference Architecture," ETSI MEC ISG, Tech. Rep., Jan. 2019.

[10] ETSI, "MEC in 5G networks," ETSI MEC ISG, Tech. Rep., Jun. 2018.

[11] N. Alliance, "Green Future Networks Sustainability Challenges and Initiatives in Mobile Networks," NGMN Alliance, Tech. Rep., Jul. 2021.

[12] T. 5. I. Association, "European Vision for the 6G Network Ecosystem," 5GIA, Tech. Rep., Jun. 2021.

[13] Z. Lai, Y. C. Hu, Y. Cui, L. Sun, N. Dai, and H. Lee, "Furion: Engineering High-Quality Immersive Virtual Reality on Today's Mobile Devices," *IEEE Trans. Mobile Comput.*, vol. 19, no. 7, pp. 1586–1602, 2020. DOI: 10.1109/TMC.2019.2913364.

[14] Y. Zhang, P. Qu, J. Cihang, and W. Zheng, "A Cloud Gaming System Based on User-Level Virtualization and Its Resource Scheduling," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 5, pp. 1239–1252, 2016. DOI: 10.1109/TPDS.2015.2433916.

[15] L. Bonati, S. D'Oro, M. Polese, S. Basagni, and T. Melodia, "Intelligence and Learning in O-RAN for Data-Driven NextG Cellular Networks," *IEEE Communications Magazine*, vol. 59, no. 10, pp. 21–27, 2021. DOI: 10.1109/MCOM.101.2001120.

[16] J. Moar, "Will Cloud Gaming change the way we play?" *White Paper, Juniper Research Ltd*, pp. 1–7, Sep. 2020.

[17] H. T. Co. "AR Insight and Application Practice White Paper." https://carrier.huawei.com/~/media/CN insight-and-application-practice-white-paper-en.pdf. (2021).

[18] E. Mills, N. Bourassa, L. Rainer, J. Mai, A. Shehabi, and N. Mills, "Toward Greener Gaming: Estimating National Energy Use and Energy Efficiency Potential," *The Computer Games Journal*, vol. 8, no. 3, pp. 157–178, 2019. DOI: 10.1007/s40869-019-00084-2. [Online]. Available: https://doi.org/10.1007/s40869-019-00084-2.

[19] Y. Li, Y. Deng, X. Tang, W. Cai, X. Liu, and G. Wang, "Cost-Efficient Server Provisioning for Cloud Gaming," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 14, no. 3s, Jun. 2018. DOI: 10.1145/3190838. [Online]. Available: https://doi.org/10.1145/3190838.

[20] Y. Deng, Y. Li, X. Tang, and W. Cai, "Server Allocation for Multiplayer Cloud Gaming," in *Proceedings of the 24th ACM International Conference on Multimedia*, ser. MM '16, Amsterdam, The Netherlands: Association for Computing Machinery, 2016, pp. 918–927. DOI: 10.1145/2964284.2964301. [Online]. Available: https://doi.org/10.1145/2964284.2964301.

[21] D. Wu, Z. Xue, and J. He, "*iCloudAccess*: Cost-Effective Streaming of Video Games From the Cloud With Low Latency," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 8, pp. 1405–1416, 2014. DOI: 10.1109/TCSVT.2014.2302543.

[22] Y. Li *et al.*, "Towards Minimizing Resource Usage With QoS Guarantee in Cloud Gaming," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 2, pp. 426–440, 2021. DOI: 10.1109/TPDS.2020.3024068.

[23] H. Chen *et al.*, "T-Gaming: A Cost-Efficient Cloud Gaming System at Scale," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 12, pp. 2849–2865, 2019. DOI: 10.1109/TPDS.2019.2922205.

[24] H.-J. Hong, D.-Y. Chen, C.-Y. Huang, K.-T. Chen, and C.-H. Hsu, "Placing Virtual Machines to Optimize Cloud Gaming Experience," *IEEE Transactions on Cloud Computing*, vol. 3, no. 1, pp. 42–53, 2015. DOI: 10.1109/TCC.2014.2338295.

[25] Y. Han, D. Guo, W. Cai, X. Wang, and V. Leung, "Virtual Machine Placement Optimization in Mobile Cloud Gaming through QoE-Oriented Resource Competition," *IEEE Transactions on Cloud Computing*, pp. 1–1, 2020. DOI: 10.1109/TCC.2020.3002023.

[26] K. Bilal and A. Erbad, "Edge computing for interactive media and video streaming," in *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*, 2017, pp. 68–73. DOI: 10.1109/FMEC.2017.7946410.

[27] L. Lin, X. Liao, H. Jin, and P. Li, "Computation Offloading Toward Edge Computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1584–1607, 2019. DOI: 10.1109/JPROC.2019.2922285.

[28] X. Zhang *et al.*, "Improving Cloud Gaming Experience through Mobile Edge Computing," *IEEE Wireless Communications*, vol. 26, no. 4, pp. 178–183, 2019. DOI: 10.1109/MWC.2019.1800440.

[29] R. D. Yates, M. Tavan, Y. Hu, and D. Raychaudhuri, "Timely cloud gaming," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, 2017, pp. 1–9. DOI: 10.1109/INFOCOM.2017.8057197.

[30] T. Braud, P. Zhou, J. Kangasharju, and P. Hui, "Multipath Computation Offloading for Mobile Augmented Reality," in *IEEE Int. Conf. Pervasive Computing and Communications (PerCom)*, 2020, pp. 1–10. DOI: 10.1109/PerCom45495.2020.9127360.

[31] J. Ren, Y. He, G. Huang, G. Yu, Y. Cai, and Z. Zhang, "An Edge-Computing Based Architecture for Mobile Augmented Reality," *IEEE Network*, vol. 33, no. 4, pp. 162–169, 2019. DOI: 10.1109/MNET.2018.1800132.

[32] Q. Liu, S. Huang, J. Opadere, and T. Han, "An Edge Network Orchestrator for Mobile Augmented Reality," in *IEEE Conf. on Computer Communications (INFOCOM)*, 2018, pp. 756–764. DOI: 10.1109/INFOCOM.2018.8486241.

[33] Y. Wang, T. Yu, and K. Sakaguchi, "Context-Based MEC Platform for Augmented-Reality Services in 5G Networks," in *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*, 2021, pp. 1–5. DOI: 10.1109/VTC2021-Fall52928.2021.9625304.

[34] D. G. Morín, P. Pérez, and A. G. Armada, "Toward the Distributed Implementation of Immersive Augmented Reality Architectures on 5G Networks," *IEEE Commun. Magazine*, vol. 60, no. 2, pp. 46–52, 2022. DOI: 10.1109/MCOM.001.2100225.

[35] E. J. Oughton and Z. Frias, "The cost, coverage and rollout implications of 5G infrastructure in Britain," *Telecommunications Policy*, vol. 42, no. 8, pp. 636–652, 2018, The implications of 5G networks: Paving the way for mobile innovation? DOI: https://doi.org/10.1016/j.telpol.2017.07.009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0308596117302781.

[36] Y. Li and M. Chen, "Software-Defined Network Function Virtualization: A Survey," *IEEE Access*, vol. 3, pp. 2542–2553, 2015.

[37] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1376–1411, 2023. DOI: 10.1109/COMST.2023.3239220.

[38] *Build resilient infrastructure, promote sustainable industrialization and foster innovation*, https://www.un.org/sustainabledevelopment/infrastructure-industrialization/, Accessed: 2024-02-21.

[39] ETSI, "Environmental Engineering (EE): Measurement method for energy efficiency of wireless access network equipment Dynamic energy performance measurement method of 5G Base Station (BS)," ETSI TS 103 786, Tech. Rep., Dec. 2020.

[40] S.-P. Chuah, C. Yuen, and N.-M. Cheung, "Cloud gaming: a green solution to massive multiplayer online games," *IEEE Wireless Communications*, vol. 21, no. 4, pp. 78–87, 2014. DOI: 10.1109/MWC.2014.6882299.

[41] H. Guan, J. Yao, Z. Qi, and R. Wang, "Energy-Efficient SLA Guarantees for Virtualized GPU in Cloud Gaming," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 9, pp. 2434–2443, 2015. DOI: 10.1109/TPDS.2014.2350499.

[42] X. Chen and G. Liu, "Energy-Efficient Task Offloading and Resource Allocation via Deep Reinforcement Learning for Augmented Reality in Mobile Edge Networks," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10 843–10 856, 2021. DOI: 10.1109/JIOT.2021.3050804.

[43] M. Mehrabi *et al.*, "Mobility- and Energy-Aware Cooperative Edge Offloading for Dependent Computation Tasks," *Network*, vol. 1, no. 2, pp. 191–214, 2021.

[44] M. Chen, W. Liu, T. Wang, A. Liu, and Z. Zeng, "Edge intelligence computing for mobile augmented reality with deep reinforcement learning approach," *Computer Networks*, vol. 195, p. 108 186, 2021.

[45] H. Wang and J. Xie, "User Preference Based Energy-Aware Mobile AR System with Edge Computing," in *IEEE Conf. on Computer Communications (INFOCOM)*, 2020, pp. 1379–1388. DOI: 10.1109/INFOCOM41043.2020.9155517.

[46] Y. Cheng, "Edge caching and computing in 5G for mobile augmented reality and haptic internet," *Computer Commun.*, vol. 158, pp. 24–31, 2020.

[47] H. Xiao, C. Xu, Y. Ma, S. Yang, L. Zhong, and G.-M. Muntean, "Edge Computing-Assisted Multimedia Service Energy Optimization based on Deep Reinforcement Learning," in *IEEE Global Communications Conf. (GLOBECOM)*, 2021. DOI: 10.1109/GLOBECOM46510.2021.9685687.

[48] J. Ahn, J. Lee, D. Niyato, and H.-S. Park, "Novel QoS-Guaranteed Orchestration Scheme for Energy-Efficient Mobile Augmented Reality Applications in Multi-Access Edge Computing," *IEEE Trans. on Vehicular Technology*, vol. 69, no. 11, pp. 13 631–13 645, 2020. DOI: 10.1109/TVT.2020.3020982.

[49] H. Huang, Q. Ye, and Y. Zhou, "Deadline-Aware Task Offloading With Partially-Observable Deep Reinforcement Learning for Multi-Access Edge Computing," *IEEE Trans. Netw. Science and Eng.*, vol. 9, no. 6, pp. 3870–3885, 2022. DOI: 10.1109/TNSE.2021.3115054.

[50] Y. Mao, Y. Luo, J. Zhang, and K. B. Letaief, "Energy harvesting small cell networks: Feasibility, deployment, and operation," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 94–101, 2015. DOI: 10.1109/MCOM.2015.7120023.

[51] K. Li, H. Zheng, and J. Wu, "Migration-based virtual machine placement in cloud systems," in *2013 IEEE 2nd International Conference on Cloud Networking (CloudNet)*, 2013, pp. 83–90.

[52] V. Farhadi *et al.*, "Service Placement and Request Scheduling for Data-Intensive Applications in Edge Clouds," *IEEE/ACM Transactions on Networking*, vol. 29, no. 2, pp. 779–792, 2021. DOI: 10.1109/TNET.2020.3048613.

[53] K. Poularakis, J. Llorca, A. M. Tulino, I. Taylor, and L. Tassiulas, "Service Placement and Request Routing in MEC Networks With Storage, Computation, and Communication Constraints," *IEEE/ACM Transactions on Networking*, pp. 1–14, 2020.

[54] D. Wang, X. Tian, H. Cui, and Z. Liu, "Reinforcement learning-based joint task offloading and migration schemes optimization in mobility-aware MEC network," *China Communications*, vol. 17, no. 8, pp. 31–44, 2020. DOI: 10.23919/JCC.2020.08.003.

[55] I. Labriji *et al.*, "Mobility Aware and Dynamic Migration of MEC Services for the Internet of Vehicles," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 570–584, 2021. DOI: 10.1109/TNSM.2021.3052808.

[56] C. Liu, F. Tang, Y. Hu, K. Li, Z. Tang, and K. Li, "Distributed Task Migration Optimization in MEC by Extending Multi-Agent Deep Reinforcement Learning Approach," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 7, pp. 1603–1614, 2021. DOI: 10.1109/TPDS.2020.3046737.

[57] R. Li *et al.*, "Energy-aware decision-making for dynamic task migration in MEC-based unmanned aerial vehicle delivery system," *Concurrency and Computation: Practice and Experience*, vol. n/a, no. n/a, e6092, 2020. DOI: https://doi.org/10.1002/cpe.6092. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpe.6092. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.6092.

[58] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, "Dynamic Service Migration in Mobile Edge Computing Based on Markov Decision Process," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1272–1288, 2019. DOI: 10.1109/TNET.2019.2916577.

[59] G. J. L. Paulraj, S. A. J. Francis, D. Peter, and I. J. Jebadurai, "Resource-aware virtual machine migration in IoT cloud," *Future Generation Computer Systems*, vol. 85, pp. 173–183, 2018. DOI: https://doi.org/10.1016/j.future.2018.03.024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X17322471.

[60] L. Gu, J. Cai, D. Zeng, Y. Zhang, H. Jin, and W. Dai, "Energy efficient task allocation and energy scheduling in green energy powered edge computing," *Future Generation Computer Systems*, vol. 95, pp. 89–99, 2019. DOI: https://doi.org/10.1016/j.future.2018.12.062. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X18310550.

[61] P. J. Braun, S. Pandi, R. Schmoll, and F. H. P. Fitzek, "On the study and deployment of mobile edge cloud for tactile internet using a 5G gaming application," in *2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC)*, Jan. 2017, pp. 154–159. DOI: 10.1109/CCNC.2017.7983098.

[62] T. Braud, A. Alhilal, and P. Hui, "Talaria: In-Engine Synchronisation for Seamless Migration of Mobile Edge Gaming Instances," in *Proceedings of the 17th International Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT '21, Virtual Event, Germany: Association for Computing Machinery, 2021, pp. 375–381. DOI: 10.1145/3485983.3494848. [Online]. Available: https://doi.org/10.1145/3485983.3494848.

[63] T. Liu, S. Ni, X. Li, Y. Zhu, L. Kong, and Y. Yang, "Deep Reinforcement Learning Based Approach for Online Service Placement and Computation Resource Allocation in Edge Computing," *IEEE Transactions on Mobile Computing*, vol. 22, no. 7, pp. 3870–3881, 2023. DOI: 10.1109/TMC.2022.3148254.

[64] M. Tang and V. W. Wong, "Deep Reinforcement Learning for Task Offloading in Mobile Edge Computing Systems," *IEEE Transactions on Mobile Computing*, vol. 21, no. 6, pp. 1985–1997, 2022. DOI: 10.1109/TMC.2020.3036871.

[65] X. Deng, J. Zhang, H. Zhang, and P. Jiang, "Deep-Reinforcement-Learning-Based Resource Allocation for Cloud Gaming via Edge Computing," *IEEE Internet of Things Journal*, vol. 10, no. 6, pp. 5364–5377, 2023. DOI: 10.1109/JIOT.2022.3222210.

[66] ETSI, "Mobile Edge Computing — A key technology towards 5G. ETSI White Paper," ETSI MEC ISG, Tech. Rep., Sep. 2015.

[67] M. Patel, B. Naughton, C. Chan, N. Sprecher, S. Abeta, A. Neal, *et al.*, "Mobile-Edge Computing introductory technical White Paper," *White paper, mobile-edge computing (MEC) industry initiative*, pp. 1089–7801, 2014.

[68] M. S. Kiryong Ha, "Openstack++ for cloudlet deployment," School of Computer Science Carnegie Mellon University Pittsburgh, Aug. 2015.

[69] J. G. Andrews *et al.*, "What Will 5G Be?" *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014. DOI: 10.1109/JSAC.2014.2328098.

[70] 3GPP, "3GPP technical specification group services and system aspects; system architecture for the 5G system," 3GPP, Tech. Rep., 2019, version TS 123 501.

[71] ETSI, "Multi-Access Edge Computing (MEC); support for network slicing," ETSI MEC ISG, Tech. Rep., Nov. 2019.

[72] A. Ksentini and P. A. Frangoudis, "Toward slicing-enabled Multi-Access Edge Computing in 5G," *IEEE Network*, vol. 34, no. 2, pp. 99–105, 2020.

[73] L. Cominardi, T. Deiss, M. Filippou, V. Sciancalepore, F. Giust, and D. Sabella, "MEC Support for Network Slicing: Status and Limitations from a Standardization Viewpoint," *IEEE Communications Standards Magazine*, vol. 4, no. 2, pp. 22–30, 2020. DOI: 10.1109/MCOMSTD.001.1900046.

[74] 3GPP, "Study on enhancement of support for Edge computing in 5G core network (5GC)," 3GPP, Tech. Rep., 2020, version SA2 TR 23.748.

[75] 3GPP, "Architecture for enabling Edge applications;" 3GPP, Tech. Rep., 2020, version SA6 TS23.558.

[76] 3GPP, "Study on application architecture for enabling Edge applications," 3GPP, Tech. Rep., 2019, version TR 23.758.

[77] ETSI, "Mobile Edge Computing (MEC); Deployment of Mobile Edge Computing in an NFV environment," ETSI MEC ISG, Tech. Rep., Feb. 2018.

[78] ETSI, "Cloud RAN and MEC: A Perfect Pairing," *ETSI MEC ISG*, no. 23, p. 25, 2018. [Online]. Available: www.etsi.org.

[79] ETSI, "MEC in an Enterprise Setting : A Solution Outline White Paper," *ETSI MEC ISG*, vol. 2, no. 30, 2018.

[80] ETSI, "Multi-Access Edge Computing (MEC); Study on MEC Support for V2X Use Cases," ETSI MEC ISG, Tech. Rep., 2018.

[81] ETSI, "MEC federation: Deployment considerations," ETSI MEC ISG, Tech. Rep., Jun. 2022.

[82] ETSI, "MEC security; Status of standards support and future evolutions," ETSI MEC ISG, Tech. Rep., Sep. 2022.

[83] ETSI, "MEC support towards Edge Native Design," ETSI MEC ISG, Tech. Rep., Jun. 2023.

[84] ETSI, "Enabling Multi-access Edge Computing in Internet-of- Things: How to deploy ETSI MEC and oneM2M," ETSI MEC ISG, Tech. Rep., Jun. 2023.

[85] S. Arora, P. A. Frangoudis, and A. Ksentini, "Exposing radio network information in a MEC-in-NFV environment: The RNISaaS concept," in *2019 IEEE Conference on Network Softwarization (NetSoft)*, 2019, pp. 306–310.

[86] ETSI, "Multi-Access Edge Computing (MEC); Radio Network Information API slicing," ETSI MEC ISG, Tech. Rep., Dec. 2019.

[87] L. Zanzi, F. Giust, and V. Sciancalepore, "M2EC: A multi-tenant resource orchestration in multi-access edge computing systems," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2018, pp. 1–6. DOI: 10.1109/WCNC.2018.8377292.

[88] F. Barbarulo, C. Puliafito, A. Virdis, and E. Mingozzi, "Extending ETSI MEC Towards Stateful Application Relocation Based on Container Migration," in *2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2022, pp. 367–376. DOI: 10.1109/WoWMoM54355.2022.00035.

[89] R. F. Castellano Gabriele Manzolini Antonio, "A Disaggregated MEC Architecture Enabling Open Services and Novel Business Models," in *IEEE Conference on Network Softwarization (Netsoft)*, Jun. 2019.

[90] E. Rojas, C. Guimarães, A. de la Oliva, C. J. Bernardos, and R. Gazda, "Beyond Multi-Access Edge Computing: Essentials to Realize a Mobile, Constrained Edge," *IEEE Communications Magazine*, vol. 62, no. 1, pp. 156–162, 2024. DOI: 10.1109/MCOM.017.2300056.

[91] A. Huang, N. Nikaein, T. Stenbock, A. Ksentini, and C. Bonnet, "Low latency MEC framework for SDN-based LTE/LTE-A networks," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6. DOI: 10.1109/ICC.2017.7996359.

[92] T. Taleb, P. A. Frangoudis, I. Benkacem, and A. Ksentini, "CDN Slicing over a Multi-Domain Edge Cloud," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2019.

[93] A. Noferi, G. Nardini, G. Stea, and A. Virdis, "Rapid prototyping and performance evaluation of ETSI MEC-based applications," *Simulation Modelling Practice and Theory*, vol. 123, p. 102 700, 2023. DOI: https://doi.org/10.1016/j.simpat.2022.102700. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569190X22001691.

[94] C. Cicconetti *et al.*, "A Prototype for QKD-secure Serverless Computing with ETSI MEC," in *2023 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2023, pp. 189–190. DOI: 10.1109/SMARTCOMP58114.2023.00043.

[95] H. A. Alameddine, S. Sharafeddine, S. Sebbah, S. Ayoubi, and C. Assi, "Dynamic Task Offloading and Scheduling for Low-Latency IoT services in Multi-Access Edge Computing," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 3, pp. 668–682, Mar. 2019. DOI: 10.1109/JSAC.2019.2894306.

[96] X. He, R. Jin, and H. Dai, "Peace: Privacy-Preserving and Cost-Efficient Task Offloading for Mobile-Edge Computing," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 1814–1824, 2020.

[97] J. Yan, S. Bi, Y. J. Zhang, and M. Tao, "Optimal Task Offloading and Resource Allocation in Mobile-Edge Computing with Inter-User Task Dependency," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 235–250, 2020.

[98] X. Meng, W. Wang, Y. Wang, V. K. N. Lau, and Z. Zhang, "Closed-Form Delay-Optimal Computation Offloading in Mobile Edge Computing Systems," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4653–4667, 2019.

[99] R. Dong, C. She, W. Hardjawana, Y. Li, and B. Vucetic, "Deep Learning for Hybrid 5G services in Mobile Edge Computing Systems: Learn From a Digital win," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4692–4707, 2019.

[100] Z. Liang, Y. Liu, T. Lok, and K. Huang, "Multiuser Computation Offloading and Downloading for Edge Computing With Virtualization," *IEEE Transactions on Wireless Communications*, vol. 18, no. 9, pp. 4298–4311, 2019.

[101] S. Jošilo and G. Dán, "Computation Offloading Scheduling for Periodic Tasks in Mobile Edge Computing," *IEEE/ACM Transactions on Networking*, pp. 1–14, 2020.

[102] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, "Computation Offloading and Resource Allocation in Wireless Cellular Networks With Mobile Edge Computing," *IEEE Transactions on Wireless Communications*, vol. 16, no. 8, pp. 4924–4938, Aug. 2017. DOI: 10.1109/TWC.2017.2703901.

[103] Y. Cheng, J. Zhang, L. Yang, C. Zhu, and H. Zhu, "Distributed Green Offloading and Power Optimization in Virtualized Small Cell Networks With Mobile Edge Computing," *IEEE Transactions on Green Communications and Networking*, vol. 4, no. 1, pp. 69–82, 2020.

[104] M. Li, F. R. Yu, P. Si, and Y. Zhang, "Energy-Efficient Machine-to-Machine (M2M) Communications in Virtualized Cellular Networks with Mobile Edge Computing (MEC)," *IEEE Transactions on Mobile Computing*, vol. 18, no. 7, pp. 1541–1555, 2019.

[105] S. Mao, S. Leng, S. Maharjan, and Y. Zhang, "Energy Efficiency and Delay Tradeoff for Wireless Powered Mobile-Edge Computing Systems With Multi-Access Schemes," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 1855–1867, 2020.

[106] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint Optimization of Radio and Computational Resources for Multicell Mobile-Edge Computing," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 1, no. 2, pp. 89–103, Jun. 2015. DOI: 10.1109/TSIPN.2015.2448520.

[107] Y. He, M. Yang, Z. He, and M. Guizani, "Computation Offloading and Resource Allocation Based on DT-MEC-Assisted Federated Learning Framework," *IEEE Transactions on Cognitive Communications and Networking*, vol. 9, no. 6, pp. 1707–1720, 2023. DOI: 10.1109/TCCN.2023.3298926.

[108] R. Zhang, C. Pan, Y. Wang, Y. Yao, and X. Li, "Federated deep reinforcement learning for multimedia task offloading and resource allocation in mec networks," *IEICE Transactions on Communications*, pp. 1–13, 2024. DOI: 10.23919/transcom.2023EBP3116.

[109] B. Li, R. Yang, L. Liu, J. Wang, N. Zhang, and M. Dong, "Robust Computation Offloading and Trajectory Optimization for Multi-Uav-Assisted MEC: A Multiagent DRL Approach," *IEEE Internet of Things Journal*, vol. 11, no. 3, pp. 4775–4786, 2024. DOI: 10.1109/JIOT.2023.3300718.

[110] D. S. Lakew, A.-T. Tran, N.-N. Dao, and S. Cho, "Intelligent Self-Pptimization for Task Offloading in LEO-MEC-Assisted Energy-Harvesting-UAV Systems," *IEEE Transactions on Network Science and Engineering*, pp. 1–14, 2024. DOI: 10.1109/TNSE.2023.3349321.

[111] Q. Tang *et al.*, "Stochastic Computation Offloading for LEO Satellite Edge Computing Networks: A Learning-Based Approach," *IEEE Internet of Things Journal*, vol. 11, no. 4, pp. 5638–5652, 2024. DOI: 10.1109/JIOT.2023.3307707.

[112] A. Ndikumana, K. K. Nguyen, and M. Cheriet, "Federated Learning Assisted Deep Q-Learning for Joint Task Offloading and Fronthaul Segment Routing in Open RAN," *IEEE Transactions on Network and Service Management*, vol. 20, no. 3, pp. 3261–3273, 2023. DOI: 10.1109/TNSM.2023.3245544.

[113] Y. Liu, L. Jiang, Q. Qi, K. Xie, and S. Xie, "Online Computation Offloading for Collaborative Space/Aerial-Aided Edge Computing Toward 6G System," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 2, pp. 2495–2505, 2024. DOI: 10.1109/TVT.2023.3312676.

[114] Q. Liu, R. Luo, H. Liang, and Q. Liu, "Energy-Efficient Joint Computation Offloading and Resource Allocation Strategy for ISAC-Aided 6G V2X Networks," *IEEE Transactions on Green Communications and Networking*, vol. 7, no. 1, pp. 413–423, 2023. DOI: 10.1109/TGCN.2023.3234263.

[115] L. Zhang *et al.*, "Digital Twin-Assisted Edge Computation Offloading in Industrial Internet of Things With NOMA," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 9, pp. 11 935–11 950, 2023. DOI: 10.1109/TVT.2023.3270859.

[116] Y. Wang, J. Fang, Y. Cheng, H. She, Y. Guo, and G. Zheng, "Cooperative End-Edge-Cloud Computing and Resource Allocation for Digital Twin Enabled 6G Industrial IoT," *IEEE Journal of Selected Topics in Signal Processing*, pp. 1–14, 2023. DOI: 10.1109/JSTSP.2023.3345154.

[117] D. Sabella and et al., "Edge computing: From standard to actual infrastructure deployment and software development," Intel, Tech. Rep., 2019.

[118] J. Martín-Pérez, L. Cominardi, C. J. Bernardos, A. de la Oliva, and A. Azcorra, "Modeling Mobile Edge Computing Deployments for Low Latency Multimedia Services," *IEEE Transactions on Broadcasting*, vol. 65, no. 2, pp. 464–474, Jun. 2019. DOI: 10.1109/TBC.2019.2901406.

[119] M. Syamkumar, P. Barford, and R. Durairajan, "Deployment Characteristics of "The Edge" in Mobile Edge Computing," in *Proceedings of the 2018 Workshop on Mobile Edge Communications*, ser. MECOMM'18, Budapest, Hungary: ACM, 2018, pp. 43–49. DOI: 10.1145/3229556.3229557. [Online]. Available: http://doi.acm.org/10.1145/3229556.3229557.

[120] M. Bouet and V. Conan, "Geo-partitioning of MEC resources," in *Proceedings of the Workshop on Mobile Edge Communications*, ser. MECOMM '17, Los Angeles, CA, USA: ACM, 2017, pp. 43–48. DOI: 10.1145/3098208.3098216. [Online]. Available: http://doi.acm.org/10.1145/3098208.3098216.

[121] P. Vitello, A. Capponi, C. Fiandrino, G. Cantelmo, and D. Kliazovich, "The Impact of Human Mobility on Edge Data Center Deployment in Urban Environments," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[122] N. Kherraf, H. A. Alameddine, S. Sharafeddine, C. M. Assi, and A. Ghrayeb, "Optimized Provisioning of Edge Computing Resources With Heterogeneous Workload in IoT Networks," *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 459–474, 2019.

[123] M. C. Filippou, D. Sabella, and V. Riccobene, "Flexible MEC service consumption through edge host zoning in 5G networks," *CoRR*, vol. abs/1903.01794, 2019. arXiv: 1903.01794. [Online]. Available: http://arxiv.org/abs/1903.01794.

[124] G. Castellano, F. Esposito, and F. Risso, "A Distributed Orchestration Algorithm for Edge Computing Resources with Guarantees," *IEEE International Conference on Computer Communications (INFOCOM 2019)*, Apr. 2019. [Online]. Available: http://par.nsf.gov/biblio/10082130.

[125] A. Virdis, G. Nardini, G. Stea, and D. Sabella, "End-to-End Performance Evaluation of MEC Deployments in 5G Scenarios," *Journal of Sensor and Actuator Networks*, vol. 9, no. 4, 2020. DOI: 10.3390/jsan9040057. [Online]. Available: https://www.mdpi.com/2224-2708/9/4/57.

[126] H. D. Chantre and N. L. Saldanha da Fonseca, "The Location Problem for the Provisioning of Protected Slices in NFV-Based MEC Infrastructure," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 7, pp. 1505–1514, 2020. DOI: 10.1109/JSAC.2020.2986869.

[127] M. C. Filippou *et al.*, "Multi-Access Edge Computing: A Comparative Analysis of 5G System Deployments and Service Consumption Locality Variants," *IEEE Communications Standards Magazine*, vol. 4, no. 2, pp. 32–39, 2020. DOI: 10.1109/MCOMSTD.001.1900034.

[128] B. Brik, P. A. Frangoudis, and A. Ksentini, "Service-Oriented MEC Applications Placement in a Federated Edge Cloud Architecture," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6. DOI: 10.1109/ICC40277.2020.9148814.

[129] F. Slim, F. Guillemin, A. Gravey, and Y. Hadjadj-Aoul, "Towards a dynamic adaptive placement of virtual network functions under ONAP," in *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Nov. 2017, pp. 210–215. DOI: 10.1109/NFV-SDN.2017.8169880.

[130] S. Salsano *et al.*, "Toward Superfluid Deployment of Virtual Functions: Exploiting Mobile Edge Computing for Video Streaming," in *2017 29th International Teletraffic Congress (ITC 29)*, vol. 2, Sep. 2017, pp. 48–53. DOI: 10.23919/ITC.2017.8065710.

[131] L. Yala, P. A. Frangoudis, and A. Ksentini, "Latency and Availability Driven VNF Placement in a MEC-NFV Environment," in *2018 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2018, pp. 1–7. DOI: 10.1109/GLOCOM.2018.8647858.

[132] K. Poularakis, J. Llorca, A. M. Tulino, I. Taylor, and L. Tassiulas, "Joint Service Placement and Request Routing in Multi-cell Mobile Edge Computing Networks," *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, Apr. 2019. DOI: 10.1109/infocom.2019.8737385. [Online]. Available: http://dx.doi.org/10.1109/INFOCOM.2019.8737385.

[133] V. Farhadi *et al.*, "Service Placement and Request Scheduling for Data-intensive Applications in Edge Clouds," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, Apr. 2019, pp. 1279–1287. DOI: 10.1109/INFOCOM.2019.8737368.

[134] S. Yang, F. Li, S. Trajanovski, X. Chen, Y. Wang, and X. Fu, "Delay-Aware Virtual Network Function Placement and Routing in Edge Clouds," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2019. DOI: 10.1109/TMC.2019.2942306.

[135] T. He, H. Khamfroush, S. Wang, T. La Porta, and S. Stein, "It's Hard to Share: Joint Service Placement and Request Scheduling in Edge Clouds with Sharable and Non-Sharable Resources," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, 2018, pp. 365–375.

[136] Q. Yuan, X. Ji, H. Tang, and W. You, "Toward Latency-Optimal Placement and Autoscaling of Monitoring Functions in MEC," *IEEE Access*, vol. 8, pp. 41 649–41 658, 2020. DOI: 10.1109/ACCESS.2020.2976858.

[137] N. Kiran, X. Liu, S. Wang, and C. Yin, "VNF Placement and Resource Allocation in SDN/NFV-Enabled MEC Networks," in *2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, 2020, pp. 1–6. DOI: 10.1109/WCNCW48565.2020.9124910.

[138] P. K. Thiruvasagam, A. Chakraborty, and C. S. R. Murthy, "Resilient and Latency-Aware Orchestration of Network Slices Using Multi-Connectivity in MEC-Enabled 5G Networks," *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 2502–2514, 2021. DOI: 10.1109/TNSM.2021.3091053.

[139] Z. Xu, W. Gong, Q. Xia, W. Liang, O. F. Rana, and G. Wu, "NFV-Enabled IoT Service Provisioning in Mobile Edge Clouds," *IEEE Transactions on Mobile Computing*, vol. 20, no. 5, pp. 1892–1906, 2021. DOI: 10.1109/TMC.2020.2972530.

[140] H.-W. Tseng, T.-T. Yang, and F.-T. Hsu, "An MEC-based VNF Placement and Scheduling Scheme for AR Application Topology," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, 2021, pp. 1–6. DOI: 10.1109/WCNC49053.2021.9417126.

[141] R. Behravesh, D. Harutyunyan, E. Coronado, and R. Riggio, "Time-Sensitive Mobile User Association and SFC Placement in MEC-Enabled 5G Networks," *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 3006–3020, 2021. DOI: 10.1109/TNSM.2021.3078814.

[142] M. Shokrnezhad, T. Taleb, and P. Dazzi, "Double Deep Q-Learning-based Path Selection and Service Placement for Latency-Sensitive Beyond 5G Applications," *IEEE Transactions on Mobile Computing*, pp. 1–14, 2023. DOI: 10.1109/TMC.2023.3301506.

[143] B. Nemeth, N. Molner, J. Martín-Pérez, C. J. Bernardos, A. de la Oliva, and B. Sonkoly, "Delay and Reliability-Constrained VNF Placement on Mobile and Volatile 5G Infrastructure," *IEEE Transactions on Mobile Computing*, vol. 21, no. 9, pp. 3150–3162, 2022. DOI: 10.1109/TMC.2021.3055426.

[144] B. P. Rimal, D. P. Van, and M. Maier, "Mobile Edge Computing Empowered Fiber-Wireless Access Networks in the 5G Era," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 192–200, Feb. 2017. DOI: 10.1109/MCOM.2017.1600156CM.

[145] A. van Kempen, T. Crivat, B. Trubert, D. Roy, and G. Pierre, "MEC-ConPaas: An Experimental Single-Board Based Mobile Edge Cloud," in *2017 5th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, Apr. 2017, pp. 17–24. DOI: 10.1109/MobileCloud.2017.17.

[146] A. A. Kherani *et al.*, "Development of MEC system for indigenous 5G Test-Bed," in *2021 International Conference on COMmunication Systems & NETworkS (COMSNETS)*, 2021, pp. 131–133. DOI: 10.1109/COMSNETS51098.2021.9352907.

[147] P. V. Wadatkar, R. G. Garroppo, and G. Nencioni, "5G-MEC Testbeds for V2X Applications," *Future Internet*, vol. 15, no. 5, 2023. DOI: 10.3390/fi15050175. [Online]. Available: https://www.mdpi.com/1999-5903/15/5/175.

[148] P. Cruz, N. Achir, and A. C. Viana, "On the Edge of the Deployment: A Survey on Multi-access Edge Computing," *ACM Comput. Surv.*, vol. 55, no. 5, Dec. 2022. DOI: 10.1145/3529758. [Online]. Available: https://doi.org/10.1145/3529758.

[149] X. Lyu *et al.*, "Selective Offloading in Mobile Edge Computing for the Green Internet of Things," *IEEE Network*, vol. 32, no. 1, pp. 54–60, Jan. 2018. DOI: 10.1109/MNET.2018.1700101.

[150] M. Jia, J. Cao, and W. Liang, "Optimal Cloudlet Placement and User to Cloudlet Allocation in Wireless Metropolitan Area Networks," *IEEE Transactions on Cloud Computing*, vol. 5, no. 4, pp. 725–737, 2017.

[151] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: Wireless Content Delivery Through Distributed Caching Helpers," *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 8402–8413, 2013.

[152] R. Bruschi, F. Davoli, P. Lago, C. Lombardo, and J. F. Pajo, "Personal Services Placement and Low-Latency Migration in Edge Computing Environments," in *2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Nov. 2018, pp. 1–6. DOI: 10.1109/NFV-SDN.2018.8725635.

[153] W. Attaoui, E. Sabir, H. Elbiaze, and M. Guizani, "VNF and CNF Placement in 5G: Recent Advances and Future Trends," *IEEE Transactions on Network and Service Management*, vol. 20, no. 4, pp. 4698–4733, 2023. DOI: 10.1109/TNSM.2023.3264005.

[154] T. Taleb and A. Ksentini, "Follow me cloud: Interworking federated clouds and distributed mobile networks," *IEEE Network*, vol. 27, no. 5, pp. 12–19, 2013.

[155] T. Taleb, A. Ksentini, and P. A. Frangoudis, "Follow-Me Cloud: When Cloud Services Follow Mobile Users," *IEEE Transactions on Cloud Computing*, vol. 7, no. 2, pp. 369–382, 2019.

[156] R. A. Addad, D. L. Cadette Dutra, M. Bagaa, T. Taleb, and H. Flinck, "Towards a Fast Service Migration in 5G," in *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*, Oct. 2018, pp. 1–6. DOI: 10.1109/CSCN.2018.8581836.

[157] A. Aissioui, A. Ksentini, A. M. Gueroui, and T. Taleb, "On Enabling 5G Automotive Systems Using Follow Me Edge-Cloud Concept," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 5302–5316, Jun. 2018. DOI: 10.1109/TVT.2018.2805369.

[158] I. Farris, T. Taleb, M. Bagaa, and H. Flick, "Optimizing service replication for mobile delay-sensitive applications in 5G edge network," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6. DOI: 10.1109/ICC.2017.7997282.

[159] I. Farris, T. Taleb, A. Iera, and H. Flinck, "Lightweight service replication for ultra-short latency applications in mobile edge networks," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6. DOI: 10.1109/ICC.2017.7996357.

[160] A. K. Sangaiah, D. V. Medhane, T. Han, M. S. Hossain, and G. Muhammad, "Enforcing Position-Based Confidentiality With Machine Learning Paradigm Through Mobile Edge Computing in Real-Time Industrial Informatics," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4189–4196, 2019.

[161] U. Fattore, M. Liebsch, B. Brik, and A. Ksentini, "AutoMEC: LSTM-based User Mobility Prediction for Service Management in Distributed MEC Resources," in *Proceedings of the 23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWiM '20, Alicante, Spain: Association for Computing Machinery, 2020, pp. 155–159. DOI: 10.1145/

3416010.3423246. [Online]. Available: https://doi.org/10.1145/3416010.3423246.

[162] S. D. A. Shah, M. A. Gregory, S. Li, R. d. R. Fontes, and L. Hou, "SDN-Based Service Mobility Management in MEC-Enabled 5G and Beyond Vehicular Networks," *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 13 425–13 442, 2022. DOI: 10.1109/JIOT.2022.3142157.

[163] I. Labriji, E. C. Strinati, E. Perraud, and F. Joly, "Dynamic Migration Strategy for Mobile Multi-Access Edge Computing Services," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, 2022, pp. 710–715. DOI: 10.1109/WCNC51071.2022.9771612.

[164] A. Mukhopadhyay, G. Iosifidis, and M. Ruffini, "Migration-Aware Network Services With Edge Computing," *IEEE Transactions on Network and Service Management*, vol. 19, no. 2, pp. 1458–1471, 2022. DOI: 10.1109/TNSM.2021.3139857.

[165] Z. Liang, Y. Liu, T.-M. Lok, and K. Huang, "Multi-Cell Mobile Edge Computing: Joint Service Migration and Resource Allocation," *IEEE Transactions on Wireless Communications*, vol. 20, no. 9, pp. 5898–5912, 2021. DOI: 10.1109/TWC.2021.3070974.

[166] F. Guo and M. Peng, "Efficient Mobility Management in Mobile Edge Computing Networks: Joint Handover and Service Migration," *IEEE Internet of Things Journal*, vol. 10, no. 20, pp. 18 237–18 247, 2023. DOI: 10.1109/JIOT.2023.3279842.

[167] Z. Ali, S. Khaf, Z. H. Abbas, G. Abbas, F. Muhammad, and S. Kim, "A Deep Learning Approach for Mobility-Aware and Energy-Efficient Resource Allocation in MEC," *IEEE Access*, vol. 8, pp. 179 530–179 546, 2020. DOI: 10.1109/ACCESS.2020.3028240.

[168] W. Wang, X. Zhou, T. Qiu, X. He, and S. Ge, "Location-Privacy-Aware Service Migration Against Inference Attacks in Multiuser MEC Systems," *IEEE Internet of Things Journal*, vol. 11, no. 1, pp. 1413–1426, 2024. DOI: 10.1109/JIOT.2023.3290145.

[169] G. A. Carella, M. Pauls, T. Magedanz, M. Cilloni, P. Bellavista, and L. Foschini, "Prototyping NFV-based Multi-Access Edge Computing in 5G ready networks with open baton," *2017 IEEE Conference on Network Softwarization: Softwarization Sustaining a Hyper-Connected World: en Route to 5G, NetSoft 2017*, 2017. DOI: 10.1109/NETSOFT.2017.8004237.

[170] B. I. Ismail *et al.*, "Evaluation of Docker as Edge computing platform," in *2015 IEEE Conference on Open Systems (ICOS)*, Aug. 2015, pp. 130–135. DOI: 10.1109/ICOS.2015.7377291.

[171] G. Avino, M. Malinverno, F. Malandrino, C. Casetti, and C. F. Chiasserini, "Characterizing Docker Overhead in Mobile Edge Computing Scenarios," in *Proceedings of the Workshop on Hot Topics in Container Networking and Networked Systems*, ser. HotConNet '17, Los Angeles, CA, USA: ACM, 2017, pp. 30–35. DOI: 10.1145/3094405.3094411. [Online]. Available: http://doi.acm.org/10.1145/3094405.3094411.

[172] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, "Dynamic Service Migration in Mobile Edge Computing Based on Markov Decision Process," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1272–1288, Jun. 2019. DOI: 10.1109/TNET.2019.2916577.

[173] T. V. Doan *et al.*, "Containers vs Virtual Machines: Choosing the Right Virtualization Technology for Mobile Edge Cloud," in *2019 IEEE 2nd 5G World Forum (5GWF)*, 2019, pp. 46–52.

[174] L. Ma, S. Yi, N. Carter, and Q. Li, "Efficient Live Migration of Edge Services Leveraging Container Layered Storage," *IEEE Transactions on Mobile Computing*, vol. 18, no. 9, pp. 2020–2033, 2019.

[175] A. Machen, S. Wang, K. K. Leung, B. J. Ko, and T. Salonidis, "Live Service Migration in Mobile Edge Clouds," *IEEE Wireless Communications*, vol. 25, no. 1, pp. 140–147, Feb. 2018. DOI: 10.1109/MWC.2017.1700011.

[176] R. Cziva, C. Anagnostopoulos, and D. P. Pezaros, "Dynamic, Latency-Optimal VNF Placement at the Network Edge," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, Apr. 2018, pp. 693–701. DOI: 10.1109/INFOCOM.2018.8486021.

[177] M. A. Hathibelagal, R. G. Garroppo, and G. Nencioni, "Experimental comparison of migration strategies for MEC-assisted 5G-V2X applications," *Computer Communications*, vol. 197, pp. 1–11, 2023. DOI: https://doi.org/10.1016/j.comcom.2022.10.009.

[178] P. V. Wadatkar, R. G. Garroppo, G. Nencioni, and M. Volpi, "Joint multi-objective MEH selection and traffic path computation in 5G-MEC systems," *Computer Networks*, vol. 240, p. 110 168, 2024. DOI: https://doi.org/10.1016/j.comnet.2023.110168. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128623006138.

[179] S. Choudhury, S. Maheshwari, I. Seskar, and D. Raychaudhuri, "ShareOn: Shared Resource Dynamic Container Migration Framework for Real-Time Support in Mobile Edge Clouds," *IEEE Access*, vol. 10, pp. 66 045–66 060, 2022. DOI: 10.1109/ACCESS.2022.3183122.

[180] K. Ha *et al.*, "Adaptive VM handoff across cloudlets," *Technical Report CMU-CS-15-113*, 2015.

[181] P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, Sep. 2017. DOI: 10.1109/COMST.2017.2682318.

[182] V. Sharma, U. Mukherji, V. Joseph, and S. Gupta, "Optimal energy management policies for energy harvesting sensor nodes," *IEEE Transactions on Wireless Communications*, vol. 9, no. 4, pp. 1326–1336, 2010.

[183] L. Huang and M. J. Neely, "Utility Optimal Scheduling in Energy-Harvesting Networks," *IEEE/ACM Transactions on Networking*, vol. 21, no. 4, pp. 1117–1130, 2013.

[184] J. Li, A. Wu, S. Chu, T. Liu, and F. Shu, "Mobile Edge Computing for Task Offloading in Small-Cell Networks via Belief Propagation," in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6.

[185] X. Zhang, W. Wu, S. Liu, and J. Wang, "An efficient computation offloading and resource allocation algorithm in RIS empowered MEC," *Computer Communications*, vol. 197, pp. 113–123, 2023. DOI: https://doi.org/10.1016/j.comcom.2022.10.012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366422004005.

[186] Z. Zabihi, A. M. Eftekhari Moghadam, and M. H. Rezvani, "Reinforcement Learning Methods for Computation Offloading: A Systematic Review," *ACM Comput. Surv.*, vol. 56, no. 1, Aug. 2023. DOI: 10.1145/3603703. [Online]. Available: https://doi.org/10.1145/3603703.

[187] B. Kar, W. Yahya, Y.-D. Lin, and A. Ali, "Offloading Using Traditional Optimization and Machine Learning in Federated Cloud–Edge–Fog Systems: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1199–1226, 2023. DOI: 10.1109/COMST.2023.3239579.

[188] F. Malandrino and C. Chiasserini, "Getting the Most Out of Your VNFs: Flexible Assignment of Service Priorities in 5G," *CoRR*, vol. abs/1904.00704, 2019. arXiv: 1904.00704. [Online]. Available: http://arxiv.org/abs/1904.00704.

[189] V. Frascolla *et al.*, "5G-MiEdge: Design, standardization and deployment of 5G phase II technologies: MEC and mmWaves joint development for Tokyo 2020 Olympic games," in *2017 IEEE Conference on Standards for Communications and Networking (CSCN)*, Oct. 2017, pp. 54–59. DOI: 10.1109/CSCN.2017.8088598.

[190] M. Hua, Y. Wang, C. Li, Y. Huang, and L. Yang, "UAV-aided Mobile Edge Computing Systems With One by One Access Scheme," *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 3, pp. 664–678, 2019.

[191] Z. Yang, C. Pan, K. Wang, and M. Shikh-Bahaei, "Energy Efficient Resource Allocation in UAV-Enabled Mobile Edge Computing Networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 9, pp. 4576–4589, 2019.

[192] P. Almasan, J. Suárez-Varela, A. Lutu, A. Cabellos-Aparicio, and P. Barlet-Ros, "Enhancing 5g radio planning with graph representations and deep learning," in *Proceedings of the 3rd ACM Workshop on 5G and Beyond Network Measurements, Modeling, and Use Cases*, ser. 5G-MeMU '23, New York, NY, USA: Association for Computing Machinery, 2023, pp. 14–20. DOI: 10.1145/3609382.3610509. [Online]. Available: https://doi.org/10.1145/3609382.3610509.

[193] L. Zanzi *et al.*, "Evolving Multi-Access Edge Computing to Support Enhanced IoT Deployments," *IEEE Communications Standards Magazine*, vol. 3, no. 2, pp. 26–34, Jun. 2019. DOI: 10.1109/MCOMSTD.2019.1800009.

[194] P. L. Ventre *et al.*, *Segment Routing: A Comprehensive Survey of Research Activities, Standardization Efforts and Implementation Results*, 2020. arXiv: 1904.03471 [cs.NI].

[195] C. Cicconetti, M. Conti, A. Passarella, and D. Sabella, "Toward Distributed Computing Environments with Serverless Solutions in Edge Systems," *IEEE Communications Magazine*, vol. 58, no. 3, pp. 40–46, 2020.

[196] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On Multi-Access Edge Computing: A Survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657–1681, Sep. 2017. DOI: 10.1109/COMST.2017.2705720.

[197] T. He, E. N. Ciftcioglu, S. Wang, and K. S. Chan, "Location Privacy in Mobile Edge Clouds: A Chaff-Based Approach," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2625–2636, 2017.

[198] X. He, J. Liu, R. Jin, and H. Dai, "Privacy-Aware Offloading in Mobile-Edge Computing," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, 2017, pp. 1–6.

[199] R. Khan, P. Kumar, D. N. K. Jayakody, and M. Liyanage, "A Survey on Security and Privacy of 5G Technologies: Potential Solutions, Recent Advancements, and Future Directions," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 196–248, 2020.

[200] S. Wang *et al.*, "Adaptive Federated Learning in Resource Constrained Edge Computing Systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[201] C. Zhang, P. Patras, and H. Haddadi, "Deep Learning in Mobile and Wireless Networking: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[202] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, Sep. 2017. DOI: 10.1109/COMST.2017.2745201.

[203] G. Vallero, D. Renga, M. Meo, and M. A. Marsan, "Greener RAN Operation Through Machine Learning," *IEEE Transactions on Network and Service Management*, vol. 16, no. 3, pp. 896–908, 2019.

[204] D. Zhang *et al.*, "Near-Optimal and Truthful Online Auction for Computation Offloading in Green Edge-Computing Systems," *IEEE Transactions on Mobile Computing*, vol. 19, no. 4, pp. 880–893, 2020.

[205] R. Ricart-Sanchez, P. Malagon, J. M. Alcaraz-Calero, and Q. Wang, "P4-netFPGA-based network slicing solution for 5G MEC architectures," in *2019 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*, 2019, pp. 1–2.

[206] M. Malinverno, J. Mangues-Bafalluy, C. E. Casetti, C. F. Chiasserini, M. Requena-Esteso, and J. Baranda, "An Edge-Based Framework for Enhanced Road Safety of Connected Cars," *IEEE Access*, vol. 8, pp. 58 018–58 031, 2020.

[207] G. Avino *et al.*, "A MEC-based Extended Virtual Sensing for Automotive Services," *IEEE Transactions on Network and Service Management*, pp. 1–1, 2019. DOI: 10.1109/TNSM.2019.2931878.

[208] Q.-H. Nguyen, M. Morold, K. David, and F. Dressler, "Car-to-Pedestrian communication with MEC-support for adaptive safety of Vulnerable Road Users," *Computer Communications*, vol. 150, pp. 83–93, 2020. DOI: https://doi.org/10.1016/j.comcom.2019.10.033. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366419304360.

[209] M. Malinverno, G. Avino, C. Casetti, C. F. Chiasserini, F. Malandrino, and S. Scarpina, *MEC-based Collision Avoidance for Vehicles and Vulnerable Users*, 2019. arXiv: 1911.05299 [cs.NI].

[210] M. Malinverno, G. Avino, C. Casetti, C. F. Chiasserini, F. Malandrino, and S. Scarpina, "Performance Analysis of C-V2I-Based Automotive Collision Avoidance," in *2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, 2018, pp. 1–9.

[211] Z. Xiao *et al.*, "Vehicular Task Offloading via Heat-Aware MEC Cooperation Using Game-Theoretic Method," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2038–2052, 2020.

[212] K. Zhang, Y. Mao, S. Leng, S. Maharjan, and Y. Zhang, "Optimal delay constrained offloading for vehicular edge computing networks," in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.

[213] Q. Hu, C. Wu, X. Zhao, X. Chen, Y. Ji, and T. Yoshinaga, "Vehicular Multi-Access Edge Computing With Licensed Sub-6 GHz, IEEE 802.11p and mmWave," *IEEE Access*, vol. 6, pp. 1995–2004, 2018. DOI: 10.1109/ACCESS.2017.2781263.

[214] R. Copeland *et al.*, "Technology assessment for mission-critical services on automotive virtual edge communicator (AVEC)," in *2018 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, Feb. 2018, pp. 1–8. DOI: 10.1109/ICIN.2018.8401630.

[215] F. Dressler, P. Handle, and C. Sommer, "Towards a Vehicular Cloud - Using Parked Vehicles As a Temporary Network and Storage Infrastructure," in *Proceedings of the 2014 ACM International Workshop on Wireless and Mobile Technologies for Smart Cities*, ser. WiMobCity '14, Philadelphia, Pennsylvania, USA: ACM, 2014, pp. 11–18. DOI: 10.1145/2633661.2633671. [Online]. Available: http://doi.acm.org/10.1145/2633661.2633671.

[216] F. Hagenauer, C. Sommer, T. Higuchi, O. Altintas, and F. Dressler, "Vehicular Micro Clouds As Virtual Edge Servers for Efficient Data Collection," in *Proceedings of the 2Nd ACM International Workshop on Smart, Autonomous, and Connected Vehicular Systems and Services*, ser. CarSys '17, Snowbird, Utah, USA: ACM, 2017, pp. 31–35. DOI: 10.1145/3131944.3133937. [Online]. Available: http://doi.acm.org/10.1145/3131944.3133937.

[217] F. Hagenauer, C. Sommer, T. Higuchi, O. Altintas, and F. Dressler, "Vehicular micro cloud in action: On gateway selection and gateway handovers," *Ad Hoc Networks*, vol. 78, pp. 73–83, 2018. DOI: https://doi.org/10.1016/j.adhoc.2018.05.014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870518302464.

[218] F. Dressler, G. S. Pannu, F. Hagenauer, M. Gerla, T. Higuchi, and O. Altintas, "Virtual Edge Computing Using Vehicular Micro Clouds," in *2019 International Conference on Computing, Networking and Communications (ICNC)*, Feb. 2019, pp. 537–541. DOI: 10.1109/ICCNC.2019.8685481.

[219] U. Montanaro, S. Fallah, M. Dianati, D. Oxtoby, T. Mizutani, and A. Mouzakitis, "Cloud-Assisted Distributed Control System Architecture for Platooning," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov. 2018, pp. 1258–1265. DOI: 10.1109/ITSC.2018.8569295.

[220] R. Huang, B. Chang, Y. Tsai, and Y. Liang, "Mobile Edge Computing-Based Vehicular Cloud of Cooperative Adaptive Driving for Platooning Autonomous Self Driving," in *2017 IEEE 7th International Symposium on Cloud and Service Computing (SC2)*, Nov. 2017, pp. 32–39. DOI: 10.1109/SC2.2017.13.

[221] A. Virdis, G. Nardini, and G. Stea, "A Framework for MEC-enabled Platooning," in *2019 IEEE Wireless Communications and Networking Conference Workshop (WCNCW)*, Apr. 2019, pp. 1–6. DOI: 10.1109/WCNCW.2019.8902910.

[222] A. Ndikumana and C. S. Hong, "Self-Driving Car Meets Multi-Access Edge Computing for Deep Learning-Based Caching," in *2019 International Conference on Information Networking (ICOIN)*, Jan. 2019, pp. 49–54. DOI: 10.1109/ICOIN.2019.8718113.

[223] A. Ndikumana, N. H. Tran, D. H. Kim, K. T. Kim, and C. S. Hong, "Deep Learning Based Caching for Self-Driving Cars in Multi-Access Edge Computing," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–16, 2020.

[224] S. Bang, M. Lee, and S. Ahn, "A Scalable VRU Protection System Based on Edge Servers," *IEEE Access*, vol. 11, pp. 97 590–97 604, 2023. DOI: 10.1109/ACCESS.2023.3312998.

[225] P. Teixeira, S. Sargento, P. Rito, M. Luís, and F. Castro, "A Sensing, Communication and Computing Approach for Vulnerable Road Users Safety," *IEEE Access*, vol. 11, pp. 4914–4930, 2023. DOI: 10.1109/ACCESS.2023.3235863.

[226] D. Sabella *et al.*, "Global MEC supporting automotive services: From multi-operator live trials to standardization," in *2021 IEEE Conference on Standards for Communications and Networking (CSCN)*, 2021, pp. 7–13. DOI: 10.1109/CSCN53733.2021.9686115.

[227] M. Karoui, V. Mannoni, B. Denis, and S. Mayrargue, "Performance Analysis of V2X-based Systems for Improved Vulnerable Road Users Safety," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 2022, pp. 3368–3373. DOI: 10.1109/ITSC55140.2022.9921841.

[228] S. Barmpounakis, G. Tsiatsios, M. Papadakis, E. Mitsianis, N. Koursioumpas, and N. Alonistioti, "Collision avoidance in 5G using MEC and NFV: The vulnerable road user safety use case," *Computer Networks*, vol. 172, p. 107 150, 2020. DOI: https://doi.org/10.1016/j.comnet.2020.107150. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128619315816.

[229] M. Emara, M. C. Filippou, and D. Sabella, "MEC-Enhanced Information Freshness for Safety-Critical C-V2X Communications," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2020, pp. 1–5. DOI: 10.1109/ICCWorkshops49005.2020.9145387.

[230] Z. Safavifar, C. Mechalikh, J. Xie, and F. Golpayegani, "Enhancing VRUs Safety Through Mobility-Aware Workload Orchestration with Trajectory Prediction using Reinforcement Learning," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, 2023, pp. 6132–6137. DOI: 10.1109/ITSC57777.2023.10421846.

[231] Y. He, F. R. Yu, N. Zhao, V. C. M. Leung, and H. Yin, "Software-Defined Networks with Mobile Edge Computing and Caching for Smart Cities: A Big Data Deep Reinforcement Learning Approach," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 31–37, Dec. 2017. DOI: 10.1109/MCOM.2017.1700246.

[232] C. Colman-Meixner *et al.*, "Deploying a Novel 5G-Enabled Architecture on City Infrastructure for Ultra-High Definition and Immersive Media Production and Broadcasting," *IEEE Transactions on Broadcasting*, vol. 65, no. 2, pp. 392–403, 2019.

[233] Y. Liu, C. Yang, L. Jiang, S. Xie, and Y. Zhang, "Intelligent Edge Computing for IoT-Based Energy Management in Smart Cities," *IEEE Network*, vol. 33, no. 2, pp. 111–117, Mar. 2019. DOI: 10.1109/MNET.2019.1800254.

[234] L. Zhao, J. Wang, J. Liu, and N. Kato, "Routing for Crowd Management in Smart Cities: A Deep Reinforcement Learning Perspective," *IEEE Communications Magazine*, vol. 57, no. 4, pp. 88–93, 2019.

[235] Y. Deng, Z. Chen, X. Yao, S. Hassan, and J. Wu, "Task Scheduling for Smart City Applications Based on Multi-Server Mobile Edge Computing," *IEEE Access*, vol. 7, pp. 14 410–14 421, 2019.

[236] M. A. Rahman, M. M. Rashid, M. S. Hossain, E. Hassanain, M. F. Alhamid, and M. Guizani, "Blockchain and IoT-Based Cognitive Edge Framework for Sharing Economy Services in a Smart City," *IEEE Access*, vol. 7, pp. 18 611–18 621, 2019.

[237] B. Wang, M. Li, X. Jin, and C. Guo, "A Reliable IoT Edge Computing Trust Management Mechanism for Smart Cities," *IEEE Access*, vol. 8, pp. 46 373–46 399, 2020.

[238] D. Wang, B. Bai, K. Lei, W. Zhao, Y. Yang, and Z. Han, "Enhancing Information Security via Physical Layer Approaches in Heterogeneous IoT With Multiple Access Mobile Edge Computing in Smart City," *IEEE Access*, vol. 7, pp. 54 508–54 521, 2019. DOI: 10.1109/ACCESS.2019.2913438.

[239] M. Gheisari, Q. Pham, M. Alazab, X. Zhang, C. Fernández-Campusano, and G. Srivastava, "ECA: An Edge Computing Architecture for Privacy-Preserving in IoT-Based Smart City," *IEEE Access*, vol. 7, pp. 155 779–155 786, 2019.

[240] H. Wang, X. Wang, X. Lan, T. Su, and L. Wan, "BSBL-Based Auxiliary Vehicle Position Analysis in Smart City Using Distributed MEC and UAV-Deployed IoT," *IEEE Internet of Things Journal*, vol. 10, no. 2, pp. 975–986, 2023. DOI: 10.1109/JIOT.2022.3204986.

[241] J. Xu, X. Liu, X. Li, L. Zhang, J. Jin, and Y. Yang, "Energy-Aware Computation Management Strategy for Smart Logistic System With MEC," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8544–8559, 2022. DOI: 10.1109/JIOT.2021.3115346.

[242] Y. Cao, X. Zhang, B. Zhou, X. Duan, D. Tian, and X. Dai, "MEC Intelligence Driven Electro-Mobility Management for Battery Switch Service," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4016–4029, 2021. DOI: 10.1109/TITS.2020.3004117.

[243] C. Cabrera, S. Svorobej, A. Palade, A. Kazmi, and S. Clarke, "MAACO: A Dynamic Service Placement Model for Smart Cities," *IEEE Transactions on Services Computing*, vol. 16, no. 1, pp. 424–437, 2023. DOI: 10.1109/TSC.2022.3143029.

[244] G. Gür *et al.*, "Integration of ICN and MEC in 5G and Beyond Networks: Mutual Benefits, Use Cases, Challenges, Standardization, and Future Research," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 1382–1412, 2022. DOI: 10.1109/OJCOMS.2022.3195125.

[245] L. U. Khan, I. Yaqoob, N. H. Tran, S. M. A. Kazmi, T. N. Dang, and C. S. Hong, "Edge-Computing-Enabled Smart Cities: A Comprehensive Survey," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10 200–10 232, 2020. DOI: 10.1109/JIOT.2020.2987070.

[246] H. Wu, Z. Zhang, C. Guan, K. Wolter, and M. Xu, "Collaborate Edge and Cloud Computing With Distributed Deep Learning for Smart City Internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8099–8110, 2020. DOI: 10.1109/JIOT.2020.2996784.

[247] L. Ale, N. Zhang, S. A. King, and J. Guardiola, "Spatio-temporal Bayesian Learning for Mobile Edge Computing Resource Planning in Smart Cities," *ACM Trans. Internet Technol.*, vol. 21, no. 3, Jun. 2021. DOI: 10.1145/3448613. [Online]. Available: https://doi.org/10.1145/3448613.

[248] S. Xu *et al.*, "RJCC: Reinforcement-Learning-Based Joint Communicational-and-Computational Resource Allocation Mechanism for Smart City IoT," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8059–8076, 2020. DOI: 10.1109/JIOT.2020.3002427.

[249] S. Zhou, C. Wei, C. Song, X. Pan, W. Chang, and L. Yang, "Short-Term Traffic Flow Prediction of the Smart City Using 5G Internet of Vehicles Based on Edge Computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 2, pp. 2229–2238, 2023. DOI: 10.1109/TITS.2022.3147845.

[250] Z. Wang, J. Hu, G. Min, Z. Zhao, and J. Wang, "Data-Augmentation-Based Cellular Traffic Prediction in Edge-Computing-Enabled Smart City," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4179–4187, 2021. DOI: 10.1109/TII.2020.3009159.

[251] Y. Zhao, K. Xu, H. Wang, B. Li, M. Qiao, and H. Shi, "MEC-Enabled Hierarchical Emotion Recognition and Perturbation-Aware Defense in Smart Cities," *IEEE Internet of Things Journal*, vol. 8, no. 23, pp. 16 933–16 945, 2021. DOI: 10.1109/JIOT.2021.3079304.

[252] H. Li *et al.*, "Intelligent Content Caching and User Association in Mobile Edge Computing Networks for Smart Cities," *IEEE Transactions on Network Science and Engineering*, vol. 11, no. 1, pp. 994–1007, 2024. DOI: 10.1109/TNSE.2023.3312369.

[253] H. Huang, K. Peng, and P. Liu, "A Privacy-aware Stackelberg Game Approach for Joint Pricing, Investment, Computation Offloading and Resource Allocation in MEC-enabled Smart Cities," in *2021 IEEE International Conference on Web Services (ICWS)*, 2021, pp. 651–656. DOI: 10.1109/ICWS53863.2021.00089.

[254] K. Peng, H. Huang, P. Liu, X. Xu, and V. C. M. Leung, "Joint Optimization of Energy Conservation and Privacy Preservation for Intelligent Task Offloading in MEC-Enabled S]mart Cities," *IEEE Transactions on Green Communications and Networking*, vol. 6, no. 3, pp. 1671–1682, 2022. DOI: 10.1109/TGCN.2022.3170146.

[255] Z. A. El Houda, B. Brik, A. Ksentini, and L. Khoukhi, "A MEC-Based Architecture to Secure IoT Applications using Federated Deep Learning," *IEEE Internet of Things Magazine*, vol. 6, no. 1, pp. 60–63, 2023. DOI: 10.1109/IOTM.001.2100238.

[256] X. Lin, J. Wu, S. Mumtaz, S. Garg, J. Li, and M. Guizani, "Blockchain-Based On-Demand Computing Resource Trading in IoV-Assisted Smart City," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 3, pp. 1373–1385, 2021. DOI: 10.1109/TETC.2020.2971831.

[257] X. Ye, M. Li, P. Si, R. Yang, Z. Wang, and Y. Zhang, "Collaborative and Intelligent Resource Optimization for Computing and Caching in IoV With Blockchain and MEC Using A3C Approach," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 2, pp. 1449–1463, 2023. DOI: 10.1109/TVT.2022.3210570.

[258] R. A. Uzcategui, A. J. De Sucre, and G. Acosta-Marum, "Wave: A tutorial," *IEEE Communications Magazine*, vol. 47, no. 5, pp. 126–133, May 2009. DOI: 10.1109/MCOM.2009.4939288.

[259] S. A. A. Shah, E. Ahmed, M. Imran, and S. Zeadally, "5G for Vehicular Communications," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 111–117, Jan. 2018. DOI: 10.1109/MCOM.2018.1700467.

[260] R. Soua, I. Turcanu, F. Adamsky, D. Fuhrer, and T. Engel, "Multi-Access Edge Computing for Vehicular Networks: A Position Paper," *2018 IEEE Globecom Workshops, GC Wkshps 2018 - Proceedings*, 2019. DOI: 10.1109/GLOCOMW.2018.8644392.

[261] M. Emara, M. C. Filippou, and D. Sabella, "MEC-Assisted End-to-End Latency Evaluations for C-V2X Communications," in *2018 European Conference on Networks and Communications (EuCNC)*, Jun. 2018, pp. 1–9. DOI: 10.1109/EuCNC.2018.8442825.

[262] 5GAA. "Toward fully connected vehicles: Edge computing for advanced automotive communications." (Feb. 2019).

[263] A. Kabil, K. Rabieh, F. Kaleem, and M. A. Azer, "Vehicle to Pedestrian Systems: Survey, Challenges and Recent Trends," *IEEE Access*, vol. 10, pp. 123 981–123 994, 2022. DOI: 10.1109/ACCESS.2022.3224772.

[264] J. Heinovski and F. Dressler, "Platoon Formation: Optimized Car to Platoon Assignment Strategies and Protocols," in *2018 IEEE Vehicular Networking Conference (VNC)*, Dec. 2018, pp. 1–8. DOI: 10.1109/VNC.2018.8628396.

[265] T. Cui, X. Fan, C. Cao, and Q. Chen, "Minimum Cost Offloading Decision Strategy for Collaborative Task Execution of Platooning Assisted by MEC," in *Communications and Networking*, X. Liu, D. Cheng, and L. Jinfeng, Eds., Cham: Springer International Publishing, 2019, pp. 104–115.

[266] C. Quadri, V. Mancuso, M. G. A. Marsan, and G. P. Rossi, "Platooning on the edge," in *Proceedings of the 23nd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWIM '20, Alicante, Spain: Association for Computing Machinery, 2020.

[267] C. Quadri, V. Mancuso, M. A. Marsan, and G. P. Rossi, "Edge-based platoon control," *Computer Communications*, vol. 181, pp. 17–31, 2022. DOI: https://doi.org/10.1016/j.comcom.2021.09.021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366421003583.

[268] S. Dabbene, C. Lehmann, C. Campolo, A. Molinaro, and F. H. P. Fitzek, "A MEC-assisted Vehicle Platooning Control through Docker Containers," in *2020 IEEE 3rd Connected and Automated Vehicles Symposium (CAVS)*, 2020, pp. 1–6. DOI: 10.1109/CAVS51000.2020.9334658.

[269] G. Nardini, A. Noferi, and G. Stea, "Platooning-as-a-service in a Multi-Operator ETSI MEC Environment," *IEEE Access*, vol. 11, pp. 60 040–60 058, 2023. DOI: 10.1109/ACCESS.2023.3286023.

[270] C. M. R. Carletti, C. Casetti, J. Härri, and F. Risso, "Platoon-Local Dynamic Map: Micro cloud support for platooning cooperative perception," in *2023 19th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2023, pp. 405–410. DOI: 10.1109/WiMob58348.2023.10187883.

[271] Y. Liu *et al.*, "Joint Communication and Computation Resource Scheduling of a UAV-Assisted Mobile Edge Computing System for Platooning Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 8435–8450, 2022. DOI: 10.1109/TITS.2021.3082539.

[272] X. Duan, Y. Zhou, D. Tian, J. Zhou, Z. Sheng, and X. Shen, "Weighted Energy-Efficiency Maximization for a UAV-Assisted Multiplatoon Mobile-Edge Computing System," *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 18 208–18 220, 2022. DOI: 10.1109/JIOT.2022.3155608.

[273] D. Zheng, Y. Chen, L. Wei, B. Jiao, and L. Hanzo, "Dynamic NOMA-Based Computation Offloading in Vehicular Platoons," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 10, pp. 13 000–13 010, 2023. DOI: 10.1109/TVT.2023.3274252.

[274] C. Chen, J. Jiang, N. Lv, and S. Li, "An Intelligent Path Planning Scheme of Autonomous Vehicles Platoon Using Deep Reinforcement Learning on Network Edge," *IEEE Access*, vol. 8, pp. 99 059–99 069, 2020. DOI: 10.1109/ACCESS.2020.2998015.

[275] Q. Liu, T. Han, J. Xie, and B. Kim, "Real-Time Dynamic Map With Crowdsourcing Vehicles in Edge Computing," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 4, pp. 2810–2820, 2023. DOI: 10.1109/TIV.2022.3214119.

[276] A. Tesei, M. Luise, P. Pagano, and J. Ferreira, "Secure Multi-access Edge Computing Assisted Maneuver Control for Autonomous Vehicles," in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, 2021, pp. 1–6. DOI: 10.1109/VTC2021-Spring51267.2021.9449087.

[277] J. Li, J. Wu, G. Xu, J. Li, X. Zheng, and A. Jolfaei, "Integrating NFV and ICN for Advanced Driver-Assistance Systems," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5861–5873, 2020. DOI: 10.1109/JIOT.2019.2953988.

[278] F. Giannone, P. A. Frangoudis, A. Ksentini, and L. Valcarenghi, "Orchestrating heterogeneous MEC-based applications for connected vehicles," *Computer Networks*, vol. 180, p. 107 402, 2020. DOI: https://doi.org/10.1016/j.comnet.2020.107402. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128620301997.

[279] X. Li, R. Song, J. Fan, M. Liu, and F.-Y. Wang, "Development and Testing of Advanced Driver Assistance Systems Through Scenario-Based Systems Engineering," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 8, pp. 3968–3973, 2023. DOI: 10.1109/TIV.2023.3297168.

[280] S. Maheshwari, W. Zhang, I. Seskar, Y. Zhang, and D. Raychaudhuri, "EdgeDrive: Supporting Advanced Driver Assistance Systems using Mobile Edge Clouds Networks," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2019, pp. 1–6. DOI: 10.1109/INFCOMW.2019.8845256.

[281] M. Wu, F. R. Yu, and P. X. Liu, "Intelligence Networking for Autonomous Driving in Beyond 5G Networks With Multi-Access Edge Computing," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 6, pp. 5853–5866, 2022. DOI: 10.1109/TVT.2022.3165172.

[282] H. Ibn-Khedher, M. Laroui, H. Moungla, H. Afifi, and E. Abd-Elrahman, "Next-Generation Edge Computing Assisted Autonomous Driving Based Artificial Intelligence Algorithms," *IEEE Access*, vol. 10, pp. 53 987–54 001, 2022. DOI: 10.1109/ACCESS.2022.3174548.

[283] J. Vyas, D. Das, and S. K. Das, "Vehicular Edge Computing Based Driver Recommendation System Using Federated Learning," in *2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, 2020, pp. 675–683. DOI: 10.1109/MASS50613.2020.00087.

[284] D. Katare, D. Perino, J. Nurmi, M. Warnier, M. Janssen, and A. Y. Ding, "A Survey on Approximate Edge AI for Energy Efficient Autonomous Driving Services," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 4, pp. 2714–2754, 2023. DOI: 10.1109/COMST.2023.3302474.

[285] B. Kar, K.-M. Shieh, Y.-C. Lai, Y.-D. Lin, and H.-W. Ferng, "Qos Violation Probability Minimization in Federating Vehicular-Fogs With Cloud and Edge Systems," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 12, pp. 13 270–13 280, 2021. DOI: 10.1109/TVT.2021.3120413.

[286] M. Gerla, "Vehicular Cloud Computing," in *2012 The 11th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, Jun. 2012, pp. 152–155. DOI: 10.1109/MedHocNet.2012.6257116.

[287] R. Yu, Y. Zhang, S. Gjessing, W. Xia, and K. Yang, "Toward Cloud-based Vehicular Networks with Efficient Resource Management," *CoRR*, vol. abs/1308.6208, 2013. arXiv: 1308.6208. [Online]. Available: http://arxiv.org/abs/1308.6208.

[288] S. Garg, A. Singh, S. Batra, N. Kumar, and L. T. Yang, "UAV-Empowered Edge Computing Environment for Cyber-Threat Detection in Smart Vehicles," *IEEE Network*, vol. 32, no. 3, pp. 42–51, May 2018. DOI: 10.1109/MNET.2018.1700286.

[289] S. Husain, A. Kunz, A. Prasad, K. Samdanis, and J. Song, "Mobile Edge Computing with network resource slicing for Internet-of-Things," in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, Feb. 2018, pp. 1–6. DOI: 10.1109/WF-IoT.2018.8355232.

[290] D. Sabella, A. Vaillant, P. Kuure, U. Rauschenbach, and F. Giust, "Mobile-Edge Computing Architecture: The role of MEC in the Internet of Things," *IEEE Consumer Electronics Magazine*, vol. 5, no. 4, pp. 84–91, Oct. 2016. DOI: 10.1109/MCE.2016.2590118.

[291] W. Yu *et al.*, "A Survey on the Edge Computing for the Internet of Things," *IEEE Access*, vol. 6, pp. 6900–6919, 2018. DOI: 10.1109/ACCESS.2017.2778504.

[292] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb, "Survey on Multi-Access Edge computing for Internet of Things Realization," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2961–2991, Apr. 2018. DOI: 10.1109/COMST.2018.2849509.

[293] Q. Pham *et al.*, "A survey of Multi-Access Edge Computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art," *IEEE Access*, vol. 8, pp. 116 974–117 017, 2020. DOI: 10.1109/ACCESS.2020.3001277.

[294] F. Cirillo, D. Gómez, L. Diez, I. E. Maestro, T. B. J. Gilbert, and R. Akhavan, "Smart City IoT Services Creation through Large Scale Collaboration," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[295] C. Vallati, A. Virdis, E. Mingozzi, and G. Stea, "Mobile-Edge Computing Come Home Connecting things in future smart homes using LTE device-to-device communications," *IEEE Consumer Electronics Magazine*, vol. 5, no. 4, pp. 77–83, Oct. 2016. DOI: 10.1109/MCE.2016.2590100.

[296] T. Taleb, S. Dutta, A. Ksentini, M. Iqbal, and H. Flinck, "Mobile Edge Computing Potential in Making Cities Smarter," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 38–43, Mar. 2017. DOI: 10.1109/MCOM.2017.1600249CM.

[297] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The Eigentrust Algorithm for Reputation Management in P2P Networks," in *Proceedings of the 12th International Conference on World Wide Web*, ser. WWW '03, Budapest, Hungary: Association for Computing Machinery, 2003, pp. 640–651. DOI: 10.1145/775152.775242. [Online]. Available: https://doi.org/10.1145/775152.775242.

[298] X. Fan, L. Liu, M. Li, and Z. Su, "GroupTrust: Dependable Trust Management," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 4, pp. 1076–1090, 2017.

[299] T. X. Tran and D. Pompili, "Adaptive Bitrate Video Caching and Processing in Mobile-Edge Computing Networks," *IEEE Transactions on Mobile Computing*, vol. 18, no. 9, pp. 1965–1978, 2019.

[300] A. Mehrabi, M. Siekkinen, and A. Ylä-Jääski, "Edge Computing Assisted Adaptive Mobile Video Streaming," *IEEE Transactions on Mobile Computing*, vol. 18, no. 4, pp. 787–800, 2019.

[301] M. Liu, F. R. Yu, Y. Teng, V. C. M. Leung, and M. Song, "Distributed Resource Allocation in Blockchain-Based Video Streaming Systems With Mobile Edge Computing," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 695–708, Jan. 2019. DOI: 10.1109/TWC.2018.2885266.

[302] Y. Hung, C. Wang, and R. Hwang, "Optimizing Social Welfare of Live Video Streaming Services in Mobile Edge Computing," *IEEE Transactions on Mobile Computing*, vol. 19, no. 4, pp. 922–934, 2020.

[303] A. Martin, R. Viola, M. Zorrilla, J. Flórez, P. Angueira, and J. Montalbán, "MEc for Fair, Reliable and Efficient Media Streaming in Mobile Networks," *IEEE Transactions on Broadcasting*, pp. 1–15, 2019.

[304] S. Yang, Y. Tseng, C. Huang, and W. Lin, "Multi-Access Edge Computing Enhanced Video Streaming: Proof-of-Concept Implementation and Prediction/QoE Models," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1888–1902, 2019.

[305] C. Ge and N. Wang, "Real-time QoE estimation of DASH-based mobile video applications through edge computing," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Apr. 2018, pp. 766–771. DOI: 10.1109/INFCOMW.2018.8406935.

[306] J. Liu, G. Shou, Y. Liu, Y. Hu, and Z. Guo, "Performance Evaluation of Integrated Multi-Access Edge Computing and Fiber-Wireless Access Networks," *IEEE Access*, vol. 6, pp. 30 269–30 279, 2018. DOI: 10.1109/ACCESS.2018.2833619.

[307] S. Mangiante, G. Klas, A. Navon, Z. GuanHua, J. Ran, and M. D. Silva, "VR is on the Edge: How to Deliver 360&Deg; Videos in Mobile Networks," in *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network*, ser. VR/AR Network '17, Los Angeles, CA, USA: ACM, 2017, pp. 30–35. DOI: 10.1145/3097895.3097901. [Online]. Available: http://doi.acm.org/10.1145/3097895.3097901.

[308] E. Bastug, M. Bennis, M. Medard, and M. Debbah, "Toward interconnected Virtual Reality: Opportunities, Challenges, and Enablers," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 110–117, Jun. 2017. DOI: 10.1109/MCOM.2017.1601089.

[309] X. Yang *et al.*, "Communication-Constrained Mobile Edge Computing Systems for Wireless Virtual Reality: Scheduling and Tradeoff," *IEEE Access*, vol. 6, pp. 16 665–16 677, 2018.

[310] Y. Sun, Z. Chen, M. Tao, and H. Liu, "Communications, Caching, and Computing for Mobile Virtual Reality: Modeling and Tradeoff," *IEEE Transactions on Communications*, vol. 67, no. 11, pp. 7573–7586, 2019.

[311] A.-E. M. Taha, N. Abu Ali, H. R. Chi, and A. Radwan, "MEC Resource Offloading for QoE-Aware HAS Video Streaming," in *ICC 2021 - IEEE International Conference on Communications*, 2021, pp. 1–5. DOI: 10.1109/ICC42927.2021.9500696.

[312] W. Shi *et al.*, "QoE Ready to Respond: A QoE-aware MEC Selection Scheme for DASH-based Adaptive Video Streaming to Mobile Users," in *Proceedings of the 29th ACM International Conference on Multimedia*, ser. MM '21, Virtual Event, China: Association for Computing Machinery, 2021, pp. 4016–4024. DOI: 10.1145/3474085.3475325. [Online]. Available: https://doi.org/10.1145/3474085.3475325.

[313] W. Liu, H. Ding, H. Zhang, and D. Yuan, "Low-Latency Oriented Resource Allocation for MEC-Assisted Adaptive Bitrate Video Streaming," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 11, pp. 14 356–14 363, 2023. DOI: 10.1109/TVT.2023.3282962.

[314] W.-Y. Chen, P.-Y. Chou, C.-Y. Wang, R.-H. Hwang, and W.-T. Chen, "Dual Pricing Optimization for Live Video Streaming in Mobile Edge Computing With Joint User Association and Resource Management," *IEEE Transactions on Mobile Computing*, vol. 22, no. 2, pp. 858–873, 2023. DOI: 10.1109/TMC.2021.3089229.

[315] X. Huang, L. He, L. Wang, and F. Li, "Towards 5G: Joint Optimization of Video Segment Caching, Transcoding and Resource Allocation for Adaptive Video Streaming in a Multi-Access Edge Computing Network," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 10, pp. 10 909–10 924, 2021. DOI: 10.1109/TVT.2021.3108152.

[316] J. Miao, S. Bai, S. Mumtaz, Q. Zhang, and J. Mu, "Utility-Oriented Optimization for Video Streaming in UAV-Aided MEC Network: A DRL Approach," *IEEE Transactions on Green Communications and Networking*, pp. 1–1, 2024. DOI: 10.1109/TGCN.2024.3352173.

[317] L. Zhang and J. Chakareski, "UAV-Assisted Edge Computing and Streaming for Wireless Virtual Reality: Analysis, Algorithm Design, and Performance Guarantees," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 3, pp. 3267–3275, 2022. DOI: 10.1109/TVT.2022.3142169.

[318] J. Du, F. R. Yu, G. Lu, J. Wang, J. Jiang, and X. Chu, "MEC-Assisted Immersive VR Video Streaming Over Terahertz Wireless Networks: A Deep Reinforcement Learning Approach," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9517–9529, 2020. DOI: 10.1109/JIOT.2020.3003449.

[319] Y. Liu, J. Liu, A. Argyriou, L. Wang, and Z. Xu, "Rendering-Aware VR Video Caching Over Multi-Cell MEC Networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 3, pp. 2728–2742, 2021. DOI: 10.1109/TVT.2021.3057684.

[320] S. Kumar, A. Franklin A, J. Jin, and Y.-N. Dong, "Seer: Learning-Based 360 Video Streaming for MEC-Equipped Cellular Networks," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 6, pp. 3308–3319, 2023. DOI: 10.1109/TNSE.2023.3257403.

[321] X. Tan, S. Wang, X. Xu, Q. Zheng, J. Yang, and S. Chen, "DACOD360: Deadline-Aware Content Delivery for 360-Degree Video Streaming Over MEC Networks," *IEEE Transactions on Multimedia*, vol. 26, pp. 4168–4182, 2024. DOI: 10.1109/TMM.2023.3321439.

[322] B. Trinh and G.-M. Muntean, "A Deep Reinforcement Learning-Based Offloading Scheme for Multi-Access Edge Computing-Supported eXtended Reality Systems," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 1, pp. 1254–1264, 2023. DOI: 10.1109/TVT.2022.3207692.

[323] Y. Goh, M. Choi, J. Jung, and J.-M. Chung, "Partial Offloading MEC Optimization Scheme using Deep Reinforcement Learning for XR Real-Time MS Devices," in *2022 IEEE International Conference on Consumer Electronics (ICCE)*, 2022, pp. 1–3. DOI: 10.1109/ICCE53296.2022.9730284.

[324] R.-J. Reifert, H. Dahrouj, and A. Sezgin, "Extended Reality via Cooperative NOMA in Hybrid Cloud/Mobile-Edge Computing Networks," *IEEE Internet of Things Journal*, pp. 1–1, 2023. DOI: 10.1109/JIOT.2023.3336393.

[325] M. Aloqaily, O. Bouachir, I. A. Ridhawi, and M. Guizani, "Realizing the Metaverse in the 6G Era with AI-Enabled Network Orchestration," *IEEE Network*, vol. 37, no. 2, pp. 78–85, 2023. DOI: 10.1109/MNET.002.2200271.

[326] Y. Jiang, J. Kang, X. Ge, D. Niyato, and Z. Xiong, "QoE Analysis and Resource Allocation for Wireless Metaverse Services," *IEEE Transactions on Communications*, vol. 71, no. 8, pp. 4735–4750, 2023. DOI: 10.1109/TCOMM.2023.3282594.

[327] D. Van Huynh, S. R. Khosravirad, A. Masaracchia, O. A. Dobre, and T. Q. Duong, "Edge Intelligence-Based Ultra-Reliable and Low-Latency Communications for Digital Twin-Enabled Metaverse," *IEEE Wireless Communications Letters*, vol. 11, no. 8, pp. 1733–1737, 2022. DOI: 10.1109/LWC.2022.3179207.

[328] H. Zhang, S. Mao, D. Niyato, and Z. Han, "Location-Dependent Augmented Reality Services in Wireless Edge-Enabled Metaverse Systems," *IEEE Open Journal of the Communications Society*, vol. 4, pp. 171–183, 2023. DOI: 10.1109/OJCOMS.2023.3234254.

[329] F. Tang, X. Chen, M. Zhao, and N. Kato, "The Roadmap of Communication and Networking in 6G for the Metaverse," *IEEE Wireless Communications*, vol. 30, no. 4, pp. 72–81, 2023. DOI: 10.1109/MWC.019.2100721.

[330] L.-H. Lee *et al.*, *All One Needs to Know about Metaverse: A Complete Survey on Technological Singularity, Virtual Ecosystem, and Research Agenda*, 2021. arXiv: 2110.05352 [cs.CY].

[331] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin, "Smart Factory of Industry 4.0: Key Technologies, Application Case, and Challenges," *IEEE Access*, vol. 6, pp. 6505–6519, 2018. DOI: 10.1109/ACCESS.2017.2783682.

[332] X. Li, J. Wan, H. Dai, M. Imran, M. Xia, and A. Celesti, "A Hybrid Computing Solution and Resource Scheduling Strategy for Edge Computing in Smart Manufacturing," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4225–4234, Jul. 2019. DOI: 10.1109/TII.2019.2899679.

[333] N. Dao, Y. Lee, S. Cho, E. Kim, K. Chung, and C. Keum, "Multi-tier Multi-Access Edge Computing: The role for the fourth industrial revolution," in *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, Oct. 2017, pp. 1280–1282. DOI: 10.1109/ICTC.2017.8190921.

[334] C. Hsu, Y. Hsu, and H. Wei, "Energy-Efficient and Reliable MEC Offloading for Heterogeneous Industrial IoT Networks," in *2019 European Conference on Networks and Communications (EuCNC)*, Jun. 2019, pp. 384–388. DOI: 10.1109/EuCNC.2019.8802020.

[335] E. E. Ugwuanyi, S. Ghosh, M. Iqbal, and T. Dagiuklas, "Reliable Resource Provisioning Using Bankers' Deadlock Avoidance Algorithm in MEC for Industrial IoT," *IEEE Access*, vol. 6, pp. 43 327–43 335, 2018.

[336] S. Luo, Y. Wen, W. Xu, and D. Puthal, "Adaptive Task Offloading Auction for Industrial CPS in Mobile Edge Computing," *IEEE Access*, vol. 7, pp. 169 055–169 065, 2019.

[337] C.-W. Hsu, Y.-L. Hsu, and H.-Y. Wei, "Energy-Efficient Edge Offloading in Heterogeneous Industrial IoT Networks for Factory of Future," *IEEE Access*, vol. 8, pp. 183 035–183 050, 2020. DOI: 10.1109/ACCESS.2020.3029253.

[338] Y. Wu, X. Zhu, J. Fei, and H. Xu, "A Novel Joint Optimization Method of Multi-Agent Task Offloading and Resource Scheduling for Mobile Inspection Service in Smart Factory," *IEEE Transactions on Vehicular Technology*, pp. 1–13, 2024. DOI: 10.1109/TVT.2024.3361492.

[339] J. Moon and J. Jeong, "Smart Manufacturing Scheduling System: DQN based on Cooperative Edge Computing," in *2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, 2021, pp. 1–8. DOI: 10.1109/IMCOM51814.2021.9377434.

[340] Z. Cao, P. Zhou, R. Li, S. Huang, and D. Wu, "Multiagent Deep Reinforcement Learning for Joint Multichannel Access and Task Offloading of Mobile-Edge Computing in Industry 4.0," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6201–6213, 2020. DOI: 10.1109/JIOT.2020.2968951.

[341] S. Massari, N. Mirizzi, G. Piro, and G. Boggia, "An Open-Source Tool Modeling the ETSI-MEC Architecture in the Industry 4.0 Context," in *2021 29th Mediterranean Conference on Control and Automation (MED)*, 2021, pp. 226–231. DOI: 10.1109/MED51440.2021.9480205.

[342] D. Borsatti, G. Davoli, W. Cerroni, and C. Raffaelli, "Enabling Industrial IoT as a Service with Multi-Access Edge Computing," *IEEE Communications Magazine*, vol. 59, no. 8, pp. 21–27, 2021. DOI: 10.1109/MCOM.001.2100006.

[343] P. Bellavista, M. Fogli, C. Giannelli, and C. Stefanelli, "Application-Aware Network Traffic Management in MEC-Integrated Industrial Environments," *Future Internet*, vol. 15, no. 2, 2023. DOI: 10.3390/fi15020042. [Online]. Available: https://www.mdpi.com/1999-5903/15/2/42.

[344] L. Nkenyereye, J. Hwang, Q.-V. Pham, and J. Song, "MEIX: Evolving Multi-Access Edge Computing for Industrial Internet-of-Things Services," *IEEE Network*, vol. 35, no. 3, pp. 147–153, 2021. DOI: 10.1109/MNET.011.2000674.

[345] C. K. M. Lee, Y. Z. Huo, S. Z. Zhang, and K. K. H. Ng, "Design of a Smart Manufacturing System With the Application of Multi-Access Edge Computing and Blockchain Technology," *IEEE Access*, vol. 8, pp. 28 659–28 667, 2020. DOI: 10.1109/ACCESS.2020.2972284.

[346] A. Abdellatif, A. Emam, C.-F. Chiasserini, A. Mohamed, A. Jaoua, and R. Ward, "Edge-based Compression and Classification for Smart Healthcare Systems: Concept, Implementation and Evaluation," *Expert Systems with Applications*, vol. 117, Sep. 2018. DOI: 10.1016/j.eswa.2018.09.019.

[347] G. Muhammad, M. F. Alhamid, and X. Long, "Computing and Processing on the Edge: Smart Pathology Detection for Connected Healthcare," *IEEE Network*, vol. 33, no. 6, pp. 44–49, 2019.

[348] M. Chen, W. Li, H. Yixue, Y. Qian, and I. Humar, "Edge cognitive computing based smart healthcare system," *Future Generation Computer Systems*, vol. 86, Apr. 2018. DOI: 10.1016/j.future.2018.03.054.

[349] A. Islam and S. Y. Shin, "BHMUS: Blockchain Based Secure Outdoor Health Monitoring Scheme Using UAV in Smart City," in *2019 7th International Conference on Information and Communication Technology (ICoICT)*, 2019, pp. 1–6.

[350] A. A. Abdellatif, A. Mohamed, C. F. Chiasserini, M. Tlili, and A. Erbad, "Edge Computing for Smart Health: Context-Aware Approaches, Opportunities, and Challenges," *IEEE Network*, vol. 33, no. 3, pp. 196–203, May 2019. DOI: 10.1109/MNET.2019.1800083.

[351] P. Pace, G. Aloi, R. Gravina, G. Caliciuri, G. Fortino, and A. Liotta, "An Edge-Based Architecture to Support Efficient Applications for Healthcare Industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 481–489, 2019.

[352] T. Muhammed, R. Mehmood, A. Albeshri, and I. Katib, "UbeHealth: A Personalized Ubiquitous Cloud and Edge-Enabled Networked Healthcare System for Smart Cities," *IEEE Access*, vol. 6, pp. 32 258–32 285, 2018. DOI: 10.1109/ACCESS.2018.2846609.

[353] L. Zhang, B. Cao, Y. Li, M. Peng, and G. Feng, "A Multi-Stage Stochastic Programming-Based Offloading Policy for Fog Enabled IoT-eHealth," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 2, pp. 411–425, 2021. DOI: 10.1109/JSAC.2020.3020659.

[354] A. Feriani, A. Refaey, and E. Hossain, "Tracking Pandemics: A MEC-Enabled IoT Ecosystem with Learning Capability," *IEEE Internet of Things Magazine*, vol. 3, no. 3, pp. 40–45, 2020. DOI: 10.1109/IOTM.0001.2000142.

[355] P. Ranaweera, M. Liyanage, and A. D. Jurcut, "Novel MEC Based Approaches for Smart Hospitals to Combat COVID-19 Pandemic," *IEEE Consumer Electronics Magazine*, vol. 10, no. 2, pp. 80–91, 2021. DOI: 10.1109/MCE.2020.3031261.

[356] C. Suraci, S. Pizzi, A. Molinaro, and G. Araniti, "MEC and D2D as Enabling Technologies for a Secure and Lightweight 6G eHealth System," *IEEE Internet of Things Journal*, vol. 9, no. 13, pp. 11 524–11 532, 2022. DOI: 10.1109/JIOT.2021.3130666.

[357] L. Zhang, X. Yuan, J. Luo, C. Feng, G. Yang, and N. Zhang, "An Adaptive Resource Allocation Approach Based on User Demand Forecasting for E-Healthcare Systems," in *2022 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2022, pp. 349–354. DOI: 10.1109/ICCWorkshops53468.2022.9814663.

[358] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "BEdgeHealth: A Decentralized Architecture for Edge-Based IoMT Networks Using Blockchain," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11 743–11 757, 2021. DOI: 10.1109/JIOT.2021.3058953.

[359] X. Yuan, H. Tian, W. Zhang, H. Zhao, Z. Zhao, and N. Zhang, "CA-PSO: A Combinatorial Auction and Improved Particle Swarm Optimization based Computation Offloading Approach for E-Healthcare," in *ICC 2022 - IEEE International Conference on Communications*, 2022, pp. 3850–3855. DOI: 10.1109/ICC45855.2022.9838733.

[360] D. Alekseeva, A. Ometov, and E. S. Lohan, "Towards the Advanced Data Processing for Medical Applications Using Task Offloading Strategy," in *2022 18th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2022, pp. 51–56. DOI: 10.1109/WiMob55322.2022.9941708.

[361] G. Ananthanarayanan *et al.*, "Real-Time Video Analytics: The Killer App for Edge Computing," *Computer*, vol. 50, no. 10, pp. 58–67, 2017. DOI: 10.1109/MC.2017.3641638.

[362] T. X. Tran, D. V. Le, G. Yue, and D. Pompili, "Cooperative Hierarchical Caching and Request Scheduling in a Cloud Radio Access Network," *IEEE Transactions on Mobile Computing*, vol. 17, no. 12, pp. 2729–2743, 2018.

[363] N. Bouten, S. Latré, J. Famaey, W. Van Leekwijck, and F. De Turck, "In-Network Quality Optimization for Adaptive Video Streaming Services," *IEEE Transactions on Multimedia*, vol. 16, no. 8, pp. 2281–2293, 2014.

[364] C. Long, Y. Cao, T. Jiang, and Q. Zhang, "Edge Computing Framework for Cooperative Video Processing in Multimedia IoT Systems," *IEEE Transactions on Multimedia*, vol. 20, no. 5, pp. 1126–1139, May 2018. DOI: 10.1109/TMM.2017.2764330.

[365] X. Jiang, F. R. Yu, T. Song, and V. C. M. Leung, "A Survey on Multi-Access Edge Computing Applied to Video Streaming: Some Research Issues and Challenges," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 871–903, 2021. DOI: 10.1109/COMST.2021.3065237.

[366] M. A. Khan *et al.*, "A Survey on Mobile Edge Computing for Video Streaming: Opportunities and Challenges," *IEEE Access*, vol. 10, pp. 120 514–120 550, 2022. DOI: 10.1109/ACCESS.2022.3220694.

[367] L. Huawei Technologies Co., "Virtual Reality/Augmented Reality White Paper," China Academy of Information and Communications Technology (CAICT), Tech. Rep., Dec. 2017.

[368] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler, "Toward Low-Latency and Ultra-Reliable Virtual Reality," *IEEE Network*, vol. 32, no. 2, pp. 78–84, Mar. 2018. DOI: 10.1109/MNET.2018.1700268.

[369] Y. Liu, J. Liu, A. Argyriou, and S. Ci, "MEC-Assisted Panoramic VR Video Streaming Over Millimeter Wave Mobile Networks," *IEEE Transactions on Multimedia*, vol. 21, no. 5, pp. 1302–1316, 2019.

[370] T. J. Chua, W. Yu, and J. Zhao, "Mobile Edge Adversarial Detection for Digital Twinning to the Metaverse: A Deep Reinforcement Learning Approach," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2023. DOI: 10.1109/TWC.2023.3298265.

[371] Y. Yao, X. Chang, L. Li, J. Liu, J. Mišić, and V. B. Mišić, "DIDs-Assisted Secure Cross-Metaverse Authentication Scheme for MEC-Enabled Metaverse," in *ICC 2023 - IEEE International Conference on Communications*, 2023, pp. 6318–6323. DOI: 10.1109/ICC45041.2023.10279761.

[372] 3GPP, "Study on Communication for Automation in Vertical Domains," 3GPP, Tech. Rep., 2018, version TR 22.804.

[373] M. Hermann, T. Pentek, and B. Otto, "Design Principles for Industries 4.0 Scenarios," in *2016 49th Hawaii International Conference on System Sciences (HICSS)*, Jan. 2016, pp. 3928–3937. DOI: 10.1109/HICSS.2016.488.

[374] P. Schulz *et al.*, "Latency Critical IoT Applications in 5G: Perspective on the Design of Radio Interface and Network Architecture," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 70–78, Feb. 2017. DOI: 10.1109/MCOM.2017.1600435CM.

[375] Y. Wu, H.-N. Dai, H. Wang, Z. Xiong, and S. Guo, "A Survey of Intelligent Network Slicing Management for Industrial IoT: Integrated Approaches for Smart Transportation, Smart Energy, and Smart Factory," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 1175–1211, 2022. DOI: 10.1109/COMST.2022.3158270.

[376] T. N. Gia, M. Jiang, A. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, "Fog Computing in Healthcare Internet of Things: A Case Study on ECG Feature Extraction," in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, Oct. 2015, pp. 356–363. DOI: 10.1109/CIT/IUCC/DASC/PICOM.2015.51.

[377] X. Li, X. Huang, C. Li, R. Yu, and L. Shu, "EdgeCare: Leveraging Edge Computing for Collaborative Data Management in Mobile Healthcare Systems," *IEEE Access*, vol. 7, pp. 22 011–22 025, 2019. DOI: 10.1109/ACCESS.2019.2898265.

[378] C. Suraci *et al.*, "The Next Generation of eHealth: A Multidisciplinary Survey," *IEEE Access*, vol. 10, pp. 134 623–134 646, 2022. DOI: 10.1109/ACCESS.2022.3231446.

[379] A. Moubayed, A. Shami, P. Heidari, A. Larabi, and R. Brunner, "Edge-Enabled V2X Service Placement for Intelligent Transportation Systems," *IEEE Transactions on Mobile Computing*, vol. 20, no. 4, pp. 1380–1392, 2021. DOI: 10.1109/TMC.2020.2965929.

[380] M. Centenaro *et al.*, "Security Considerations on 5G-Enabled Back-Situation Awareness for CCAM," in *2020 IEEE 3rd 5G World Forum (5GWF)*, 2020, pp. 245–250. DOI: 10.1109/5GWF49715.2020.9221064.

[381] P. Lang, D. Tian, X. Duan, J. Zhou, Z. Sheng, and V. C. M. Leung, "Cooperative Computation Offloading in Blockchain-Based Vehicular Edge Computing Networks," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 783–798, 2022. DOI: 10.1109/TIV.2022.3190308.

[382] F. Giannone, P. A. Frangoudis, A. Ksentini, and L. Valcarenghi, "Orchestrating heterogeneous MEC-based applications for connected vehicles," *Computer Networks*, vol. 180, p. 107 402, 2020. DOI: https://doi.org/10.1016/j.comnet.2020.107402. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128620301997.

[383] A. Renda *et al.*, "Federated Learning of Explainable AI Models in 6G Systems: Towards Secure and Automated Vehicle Networking," *Information*, vol. 13, no. 8, 2022. DOI: 10.3390/info13080395. [Online]. Available: https://www.mdpi.com/2078-2489/13/8/395.

[384] B. Coll-Perales *et al.*, "End-to-End Latency of V2N2V Communications under Different 5G and Computing Deployments in Multi-MNO Scenarios," in *2022 IEEE 33rd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2022, pp. 622–627. DOI: 10.1109/PIMRC54779.2022.9978114.

[385] A. Chiha, B. Denis, S. Verbrugge, and D. Colle, "Techno-Economic Analysis of MEC Clustering Models for Seamless CCAM Service Provision," *IEEE Communications Magazine*, vol. 61, no. 2, pp. 32–37, 2023. DOI: 10.1109/MCOM.001.2200299.

[386] F. Giust *et al.*, "Multi-Access Edge Computing: The Driver Behind the Wheel of 5G-Connected Cars," *IEEE Communications Standards Magazine*, vol. 2, no. 3, pp. 66–73, Sep. 2018. DOI: 10.1109/MCOMSTD.2018.1800013.

[387] D. Grewe, M. Wagner, M. Arumaithurai, I. Psaras, and D. Kutscher, "Information-Centric Mobile Edge Computing for Connected Vehicle Environments: Challenges and Research Directions," in *Proceedings of the Workshop on Mobile Edge Communications*, ser. MECOMM '17, Los Angeles, CA, USA: ACM, 2017, pp. 7–12. DOI: 10.1145/3098208.3098210. [Online]. Available: http://doi.acm.org/10.1145/3098208.3098210.

[388] A. Alalewi, I. Dayoub, and S. Cherkaoui, "On 5G-V2X Use Cases and Enabling Technologies: A Comprehensive Survey," *IEEE Access*, vol. 9, pp. 107 710–107 737, 2021. DOI: 10.1109/ACCESS.2021.3100472.

[389] S. Hakak *et al.*, "Autonomous vehicles in 5G and beyond: A survey," *Vehicular Communications*, vol. 39, p. 100 551, 2023. DOI: https://doi.org/10.1016/j.vehcom.2022.100551. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214209622000985.

[390] W. Zhang, L. Li, N. Zhang, T. Han, and S. Wang, "Air-Ground Integrated Mobile Edge Networks: A Survey," *IEEE Access*, vol. 8, pp. 125 998–126 018, 2020. DOI: 10.1109/ACCESS.2020.3008168.

[391] O-RAN ALLIANCE, "O-RAN Towards an Open and Smart RAN White Paper," O-RAN ALLIANCE, Tech. Rep., Oct. 2019.

[392] T.-H. Chao, J.-H. Wu, Y. Chiang, and H.-Y. Wei, "5g edge computing experiments with intelligent resource allocation for multi-application video analytics," in *2021 30th Wireless and Optical Communications Conference (WOCC)*, 2021, pp. 80–84. DOI: 10.1109/WOCC53213.2021.9603242.

[393] S. Kumar, N. Wang, Y. Rahulan, and B. Evans, "Edge computing-based layered video streaming over integrated satellite and terrestrial 5g networks," *IEEE Access*, vol. 10, pp. 19 971–19 985, 2022. DOI: 10.1109/ACCESS.2022.3151998.

[394] T. Qiu, J. Chi, X. Zhou, Z. Ning, M. Atiquzzaman, and D. O. Wu, "Edge Computing in Industrial Internet of Things: Architecture, Advances and Challenges," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2020. DOI: 10.1109/COMST.2020.3009103.

[395] 5G Alliance for Connected Industries and Automation, "5G for Connected Industries and Automation," 5G ACIA, Tech. Rep., Feb. 2019.

[396] 5G Alliance for Connected Industries and Automation, "5G for automation in industry: Primary use cases, functions and service requirements," 5G ACIA, Tech. Rep., Jul. 2019.

[397] K. Antevski, C. J. Bernardos, L. Cominardi, A. de la Oliva, and A. Mourad, "On the integration of NFV and MEC technologies: Architecture analysis and benefits for edge robotics," *Computer Networks*, vol. 175, p. 107 274, 2020. DOI: https://doi.org/10.1016/j.comnet.2020.107274. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128620300797.

[398] K. Antevski, M. Groshev, G. Baldoni, and C. J. Bernardos, *DLT federation for Edge robotics*, 2020. arXiv: 2010.01977 [cs.NI].

[399] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of Edge Computing and Deep Learning: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2020.

[400] T. K. Rodrigues, K. Suto, H. Nishiyama, J. Liu, and N. Kato, "Machine Learning Meets Computation and Communication Control in Evolving Edge and Cloud: Challenges and Future Perspective," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 38–67, 2020.

[401] L. Cominardi, L. M. Contreras, C. J. Bcrnardos, and I. Berberana, "Understanding QoS Applicability in 5G Transport Networks," in *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, Jun. 2018, pp. 1–5. DOI: 10.1109/BMSB.2018.8436847.

[402] ITU-T, "Consideration on 5G transport network reference architecture and bandwidth requirements," International Telecommunication Union - Telecommunication Standardization Sector (ITU-T), Study Group 15 Contribution 0462, Feb. 2017.

[403] *State of digital communication*, https://etno.eu, Accessed: 2024-02-21.

[404] 3GPP, "Service requirements for next generation new services and markets," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 22.261, Sep. 2018, version 16.5.0.

[405] N. Khalid and O. B. Akan, "Experimental Throughput Analysis of Low-THz MIMO Communication Channel in 5G Wireless Networks," *IEEE Wireless Communications Letters*, vol. 5, no. 6, pp. 616–619, Dec. 2016. DOI: 10.1109/LWC.2016.2606392.

[406] Z. Amjad, A. Sikora, B. Hilt, and J. Lauffenburger, "Low Latency V2X Applications and Network Requirements: Performance Evaluation," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 220–225.

[407] H. Guo, J. Liu, and J. Zhang, "Efficient Computation Offloading for Multi-Access Edge Computing in 5G HetNets," in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6.

[408] W. Zhang, B. Han, and P. Hui, "On the Networking Challenges of Mobile Augmented Reality," in *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network*, ser. VR/AR Network '17, Los Angeles, CA, USA: ACM, 2017, pp. 24–29. DOI: 10.1145/3097895.3097900. [Online]. Available: http://doi.acm.org/10.1145/3097895.3097900.

[409] V. Herminghaus and A. Scriba, *Storage Management in Data Centers: Understanding, Exploiting, Tuning, and Troubleshooting Veritas Storage Foundation*. Springer Science & Business Media, 2009.

[410] T. Zhang, L. Linguaglossa, P. Giaccone, L. Iannone, and J. Roberts, *Performance Benchmarking of State-of-the-Art Software Switches for NFV*, 2020. arXiv: 2003.13489 [cs.NI].

[411] L. Kleinrock, "Time-shared Systems: A Theoretical Treatment," *J. ACM*, vol. 14, no. 2, pp. 242–261, Apr. 1967. DOI: 10.1145/321386.321388. [Online]. Available: http://doi.acm.org/10.1145/321386.321388.

[412] F. Malandrino, C. Chiasserini, G. Avino, M. Malinverno, and S. Kirkpatrick, "From Megabits to CPU Ticks: Enriching a Demand Trace in the Age of MEC," *IEEE Transactions on Big Data*, vol. 6, no. 1, pp. 43–50, 2020.

[413] Q. Wu, G. Y. Li, W. Chen, D. W. K. Ng, and R. Schober, "An Overview of Sustainable Green 5G Networks," *IEEE Wireless Communications*, vol. 24, no. 4, pp. 72–80, 2017. DOI: 10.1109/MWC.2017.1600343.

[414] T. Huang, W. Yang, J. Wu, J. Ma, X. Zhang, and D. Zhang, "A Survey on Green 6G Network: Architecture and Technologies," *IEEE Access*, vol. 7, pp. 175 758–175 768, 2019. DOI: 10.1109/ACCESS.2019.2957648.

[415] T. Kämäräinen, Y. Shan, M. Siekkinen, and A. Ylä-Jääski, "Virtual machines vs. containers in cloud gaming systems," in *2015 International Workshop on Network and Systems Support for Games (NetGames)*, 2015, pp. 1–6. DOI: 10.1109/NetGames.2015.7382987.

[416] ITU-T, "Consideration on 5G transport network reference architecture and bandwidth requirements," ITU-T Study Group, Tech. Rep., Feb. 2018.

[417] W. Li *et al.*, "On Enabling Sustainable Edge Computing with Renewable Energy Resources," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 94–101, 2018. DOI: 10.1109/MCOM.2018.1700888.

[418] D. Renga and M. Meo, "Dimensioning Renewable Energy Systems to Power Mobile Networks," *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 2, pp. 366–380, 2019. DOI: 10.1109/TGCN.2019.2892200.

[419] *Wind power production estimation and forecast on Belgian grid (near real-time)*, White paper, Elia Transmission Belgium SA, https://opendata.elia.be/explore/dataset/ods086/information/, [Accessed: 2022-03-30].

[420] *Sun power production estimation and forecast on belgian grid (near real-time)*, White paper, Elia Transmission Belgium SA, https://opendata.elia.be/explore/dataset/ods087/information/, [Accessed: 2022-03-30].

[421] D. Finkel, M. Claypool, S. Jaffe, T. Nguyen, and B. Stephen, "Assignment of games to servers in the OnLive cloud game system," in *2014 13th Annual Workshop on Network and Systems Support for Games*, 2014, pp. 1–3. DOI: 10.1109/NetGames.2014.7008958.

[422] K.-T. Chen, Y.-C. Chang, H.-J. Hsu, D.-Y. Chen, C.-Y. Huang, and C.-H. Hsu, "On the Quality of Service of Cloud Gaming Systems," *IEEE Transactions on Multimedia*, vol. 16, no. 2, pp. 480–495, 2014. DOI: 10.1109/TMM.2013.2291532.

[423] E. Mills, N. Bourassa, L. Rainer, J. Mai, A. Shehabi, and N. Mills, *Energy consumption of cloud games*, Data from work [18], https://docs.google.com/spreadsheets/d/134WlGZK3tuXFnh3igJz-L3nmpBJzXg3PznGXZagtbm4, [Accessed: 2022-03-30].

[424] E. Mills, N. Bourassa, L. Rainer, J. Mai, A. Shehabi, and N. Mills, *Supplemental information for the article entitled "Toward Greener Gaming: Estimating National Energy Use and Energy Efficiency Potential"*, Data from work [18], https://docs.google.com/spreadsheets/d/1ZXLwHuWodc6EsROfgXn0gbDkYnSHB_StEGyA36GiklM, [Accessed: 2022-03-30].

[425] S. Martello, "Knapsack Problems: Algorithms and Computer Implementations," *Wiley-Interscience series in discrete mathematics and optimiza tion*, 1990. [Online]. Available: https://ci.nii.ac.jp/naid/20000416220/en/.

[426] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey, "An analysis of approximations for maximizing submodular set functions—i," in *Mathematical Programming*, M. L. Balinski and A. J. Hoffman, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1978, pp. 265–294. DOI: 10.1007/BF01588971. [Online]. Available: https://doi.org/10.1007/BF01588971.

[427] *NVIDIA RTX Blade Server*, Data Sheet, NVIDIA Corporation, https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/cloud-gaming-server/geforce-now-rtx-server-gaming-datasheet.pdf, [Accessed: 2022-03-30].

[428] *Google Stadia Website*, Website, Google Stadia, https://stadia.google.com, [Accessed: 2022-03-30].

[429] C. E. Clark, "The PERT model for the distribution of an activity time," *Operations Research*, vol. 10, no. 3, pp. 405–406, 1962.

[430] "3GPP TR 26.928 Extended Reality (XR) in 5G," Tech. Rep. Dec. 2020.

[431] S. Rodriguez. "Crafting a Market for for independent XR." https://xnquebec.co/pdf/Etude_Distributi
().

[432] M. Latvaho, K. Leppänen, F. Clazzer, and A. Munari, *Key drivers and research challenges for 6G ubiquitous wireless intelligence* (6G research visions). Oulu, Finland: University of Oulu, 2019.

[433] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *Journal of the Operational Research society*, vol. 49, no. 3, pp. 237–252, 1998.

[434] R. Buyya and S. Srirama, *Fog and Edge Computing: Principles and Paradigms* (Wiley Series on Parallel and Distributed Computing). Wiley, 2019.

[435] J. Gil Herrera and J. F. Botero, "Resource Allocation in NFV: A Comprehensive Survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 518–532, Sep. 2016. DOI: 10.1109/TNSM.2016.2598420.

[436] V. Mnih *et al.*, "Asynchronous Methods for Deep Reinforcement Learning," *CoRR*, vol. abs/1602.01783, 2016. arXiv: 1602.01783. [Online]. Available: http://arxiv.org/abs/1602.01783.