Computer Networks 195 (2021) 108177

Contents lists available at ScienceDirect

Computer Networks

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

Survey paper

Task offloading in Edge and Cloud Computing: A survey on mathematical, artificial intelligence and control theory solutions

Firdose Saeik [a], Marios Avgeris [b], Dimitrios Spatharakis [b], Nina Santi [c], Dimitrios Dechouniotis [b], John Violos [a], Aris Leivadeas [a,*], Nikolaos Athanasopoulos [d], Nathalie Mitton [c], Symeon Papavassiliou [b]

[a] Department of Software and Information Technology Engineering, École de Technologie Supérieure, Montréal, Canada
[b] Department of Electrical and Computer Engineering, National Technical University of Athens, Greece
[c] INRIA, France
[d] School of Electronics, Electrical Engineering and Computer Science, Queen's University, Belfast, UK

ARTICLE INFO

Keywords:
Edge Computing
Task offloading
Resource allocation
Control theory
Mathematical optimization
Artificial intelligence

ABSTRACT

Next generation communication networks are expected to accommodate a high number of new and resource-voracious applications that can be offered to a large range of end users. Even though end devices are becoming more powerful, the available local resources cannot cope with the requirements of these applications. This has created a new challenge called task offloading, where computation intensive tasks need to be offloaded to more resource powerful remote devices. Naturally, the Cloud Computing is a well-tested infrastructure that can facilitate the task offloading. However, Cloud Computing as a centralized and distant infrastructure creates significant communication delays that cannot satisfy the requirements of the emerging delay-sensitive applications. To this end, the concept of Edge Computing has been proposed, where the Cloud Computing capabilities are repositioned closer to the end devices at the edge of the network. This paper provides a detailed survey of how the Edge and/or Cloud can be combined together to facilitate the task offloading problem. Particular emphasis is given on the mathematical, artificial intelligence and control theory optimization approaches that can be used to satisfy the various objectives, constraints and dynamic conditions of this end-to-end application execution approach. The survey concludes with identifying open challenges and future directions of the problem at hand.

1. Introduction

Wireless communications have come a long way over the last 40 years allowing a plethora of new applications and services to proliferate. This wireless growth has revolutionized the way humans and machines interact with each other and between them. Specifically, as wireless technologies evolve, the data rate, mobility, coverage and spectral efficiency rapidly increase [1], permitting radical changes on the grounds of our society and our personal communication. At the same time, with the advent of the Internet of Things (IoT) and emergent applications such as Virtual Reality (VR) and driverless cars, the demand for wireless communications with even higher-speeds and ubiquitous connectivity becomes a necessity that requires more efficient wireless communication systems.

5G is an exemplary wireless communication system that tries to minimize the gap between the new emergent applications and their high-performance requirements. Specifically, 5G promises the support of increased bandwidth and connection density, as well as low-latency communication, with the induction of the enhanced mobile broadband (eMMB), the massive machine-type communication (mMTC) and the ultra-reliable low latency communication (uRLLC) services [2]. However, even though the performance of the wireless access networks continues to increase, allowing the support of new and more intelligent applications, the end devices cannot always cope with the strict computational requirements of these resource voracious applications.

Inevitably, the answer to where we can find an increased availability of computational resources, accompanied with the necessary

* Corresponding author.
E-mail addresses: firdose.saeik.1@ens.etsmtl.ca (F. Saeik), mavgeris@netmode.ntua.gr (M. Avgeris), dspatharakis@netmode.ntua.gr (D. Spatharakis), nina.santi@inria.fr (N. Santi), ddechou@netmode.ntua.gr (D. Dechouniotis), violos@mail.ntua.gr (J. Violos), aris.leivadeas@etsmtl.ca (A. Leivadeas), n.athanasopoulos@qub.ac.uk (N. Athanasopoulos), nathalie.mitton@inria.fr (N. Mitton), papavass@mail.ntua.gr (S. Papavassiliou).

https://doi.org/10.1016/j.comnet.2021.108177
Received 19 October 2020; Received in revised form 24 January 2021; Accepted 14 May 2021
Available online 18 May 2021

reliability to offer a seamless communication for the wireless applications, always lies around the Cloud. The Cloud is a well tested and used solution that can extend the resource capabilities of the end devices with powerful data center topologies. Besides, Cloud is well equipped with the appropriate automation tools and platforms in order to offer the necessary transparency to the end devices, while hiding the complexity and the logistic details of this resource extension.

Hence, the practice of offloading computation intensive tasks of resource-intensive applications from the end devices to a centralized Cloud infrastructure, is a well explored solution [3][4]. Nonetheless, as the focus of new applications turned towards high throughput and low latency communications, the Cloud started to expose its inherent limitations. The long distance between the end devices and the Cloud infrastructure, the use of a best-effort and unreliable intermittent transport network, the cost of traversing the backhaul network and the increased security surface throughout this long communication path, created the need for alternative solutions.

There is no question, that these substitute solutions should introduce a more distributed infrastructure that will enhance the local efficiency by bringing Cloud-like capabilities closer to the end devices, at the edge of the network. This is exactly how the term Edge Computing was coined. Even though, multiple flavors of the Edge Computing exist (e.g., Fog Computing, Mobile Cloud Computing, Cloudlet, Mobile Edge Computing), they all agree that additional and existing computational and networking resources at the edge of the network should be inserted and regrouped.

This new infrastructural component that creates an additional resource layer between the end devices and the Cloud, is able to reduce the increased bandwidth consumption at the backhaul, transport and Cloud networks and also reduce the communication delay and support applications with real time requirements. In particular, end devices are now capable of offloading their resource-intensive tasks to a nearby Edge device, thus minimizing the overall execution time without adding excessive communication paths towards a distant Cloud infrastructure. This solution, called task offloading, allows enhancing the user's experience by providing lower latency, better reliability and improved energy efficiency for battery-powered devices.

Even though the notion of Edge Computing exists for almost a decade, the problem of task offloading has only recently started to be investigated. Nonetheless, it has gained a lot of attention from the industry and the academic community, leading to the publication of many scientific and research papers over the last couple of years. A great effort has also been made to classify and categorize the different types of task offloading by a number of surveys and tutorials.

These surveys focused on multiple aspects such as architecture [5–7], resource allocation [8,9], communication [8,10,11], caching [10], mobility management [6,10,12], integration with wireless, IoT and 5G technologies [5,8,13,14], decision on task offloading [6,11], application partitioning [12,15], application models [8,12,16,17], application scenarios [5,8,10,15,18,19] and algorithms [11,12,17].

In this survey paper, we also attempt to study the task offloading problem, emphasizing, however, on novel algorithmic and control approaches. Thus, in contrast with recent surveys on task offloading, our contribution is twofold; firstly, we provide a comprehensive survey of task offloading within three subfields: (i) Optimization algorithms (ii) Artificial Intelligence techniques and (iii) Control theory; secondly, a categorization of the above techniques is provided based on their objective function, the granularity level, the use of the Edge and/or Cloud infrastructures and the incorporation of mobility in the overall solution, depending on the type of the edge devices.

This paper is organized as follows. Section 2 presents an overview of the various computing paradigms and relevant technologies evolved in the last decade, along with some potential use cases for task offloading of interactive applications. Following, Section 2.2.6 formally defines the task offloading problem along with the challenges involved. Section 4 covers different task offloading solutions that have been recently proposed, emphasizing on the mathematical models, optimization techniques, machine learning algorithms and control theory approaches. Section 5 presents the open challenges. Finally, we conclude and provide suggestions for future work in Section 6.

2. Computing paradigms: Overview & use cases

As already discussed, over the last two decades Cloud Computing has been the dominant service delivery paradigm. However, modern applications come with strict requirements which cannot be met via execution in remote Cloud resources (e.g., ultra low delay). Thus, the current trend of resource provisioning is to augment the edge of the network with computing capabilities. Towards this direction, the emerging service model of Edge Computing promises to mitigate the limitations of Cloud Computing. To clarify the ambiguity behind the terminology and architectures used in the literature, this section provides the fundamental elements of the various modern computing infrastructures such as Cloud Computing, Mobile Cloud Computing, Mobile Edge Computing and Fog Computing. Furthermore, emerging use cases concerning task offloading at the Edge and/or Cloud are presented.

2.1. Modern computing paradigms

2.1.1. Cloud computing

Cloud Computing has revolutionized the Internet and completely transformed the way that applications, software and resources are offered to the end users. According to NIST [20], Cloud Computing is defined as "a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction". The Cloud paradigm brings unique benefits. In particular, with this model, computing resources are offered to the end users on demand, in a self-service fashion, independent of the type of the device and the location of the user. Furthermore, the computational and network resources available at the Cloud can be shared and dynamically scaled. This is achieved by adopting virtualization as the enabling technology of the Cloud, allowing the resources to be allocated and released with minimal interaction, while users pay for the service they consume according to its usage [21].

Despite bringing numerous advantages, Cloud Computing poses some serious limitations. These limitations, although they exist since the beginning of the Cloud, they did not surface until recently. The reason is that new communication technologies, new applications and services have increased the data volume generated and at the same time also increased the demands for low latency communications. Hence, offering Cloud Computing resources in a centralized manner far away from the users, can create serious delay bottlenecks. This delay can be disastrous for mission critical applications, such as health related applications, or time-critical applications like real-time control in manufacturing. Another disadvantage is that forwarding the traffic from the end devices to the Cloud, usually involves traversing the core Internet. This can create three serious problems. Firstly, sending hundreds of TB of data from the devices to the Cloud can certainly create traffic hotspots in the Internet infrastructure, which can further affect the communication delay. Secondly, the existence of various different networks and administrative domains between the Cloud and the front-end devices, can create an unstable and intermittent network connectivity. Finally, the data, before being sent to the Cloud, probably have to traverse a backhaul network (e.g., cellular and satellite). This backhaul network may be costly and lossy, creating additional problems to this end-to-end communication. Of course, there may be additional limitations, regarding for example the security aspects of the communication, since this end device-to-Cloud communication increases the surface of threat. However, in the particular survey we emphasize only on the networking and data processing limitations.

2.1.2. Mobile cloud computing

Apparently, Cloud Computing can be used for offloading tasks from mobile devices to a more powerful infrastructure. This approach created the notion of Mobile Cloud Computing (MCC). A Mobile Cloud is defined as a mobile device that can execute a resource-intensive application on a distant high-performance compute server or compute cluster and support thin client user interactions with the application over the Internet [22–25]. MCC can be thus described as the integration of mobile devices with Cloud Computing technology. It offers computing, storage, services and applications over the Internet and the typical advantages found in a Cloud Computing environment such as cost reduction and resource flexibility. In addition, MCC can potentially save energy for mobile users by offloading high-energy consuming applications to the Cloud [21].

However, such an approach still carries the typical Cloud limitations presented above. Thus, the concept of MCC can be modified to offer the necessary Cloud resources closer to the mobile devices. This new flavor of MCC is called Cloudlet [26] and it allows the mobile devices to offload their workload to a local "mini cloud", consisting of multiple multi-core hardware equipment directly connected to an Access Point (AP) or Base Station (BS). Therefore, Cloudlet can be seen as a trusted, resource-rich computer or computer cluster, which is connected to the Internet and is available for use from mobile devices in proximity. Due to the sheer proximity of Cloudlet, sharp interactive response for immersive applications that magnify human cognition is much easier to attain. Instead of depending on a remote server, a mobile user instantiates a "Cloudlet" on the local network and uses it via a wireless LAN. These proposed Cloudlets can be placed in common areas such as railway stations, restaurants and coffee shops, so that mobile devices could connect to them and act as a thin client. This opposes to the use of a centralized Cloud server that would raise issues of latency and bandwidth.

2.1.3. Fog computing

Fog Computing is another approach for expanding the Cloud Computing concept to the edge of the network, thus enabling a new range of apps and services [27,28]. Fog Computing was the first industry initiative to explicitly define an architecture for applying utility Cloud at the edge of the network, and was standardized by the OpenFog consortium [29]. Specifically, the term Fog Computing was coined by Cisco in 2012 and is defined as "the process of extending Cloud Computing capabilities at the edge of the network. Fog incorporates computing, storage and network resources close to the IoT layer to facilitate the data processing". [27,30]. From the previous definition it becomes evident that Fog Computing was introduced in order to facilitate the monitoring, control and analysis of IoT devices in real time, removing the long communication delay between the IoT devices and the central analytics application servers in a remote Cloud.

Hence, Fog Computing expands Cloud Computing by installing localized computing facilities at the user's premises, delivering Cloud data to mobile users with fast local connections. The aim is to process in part workload and services locally on Fog devices (such as hardened routers, switches, IP video cameras), rather than being transmitted to the Cloud [28]. As such, Fog Computing introduces an intermediate infrastructure layer between mobile users and Cloud, in order to support low-latency and high-speed services. Moreover, Fog Computing can support and promote applications that do not suit the Cloud [31], such as (i) applications involving very low and consistent latency, (ii) geographically distributed systems such as pipeline control and sensor networks (iii) mobile applications like smart connected vehicles and (iv) large-scale adaptive control systems, such as smart energy delivery and smart traffic lights. As such, in the literature, the typical applications usually combined with the Fog Computing are mostly IoT related [32–34], cache networks [35] and immersive media services (AR/VR, a 360-degree video and free-viewpoint video) applications [36].

2.1.4. Edge computing

Fog Computing has managed to bypass many of the limitations of Cloud Computing, increasing the performance of IoT and mobile applications in terms of task offloading. However, stringent requirements such as ultra-low latency, user experience, stability and high reliability have created the need for even higher localized information near end users. Thus, Edge Computing is another similar concept, that can be defined as a network layer encompassing end devices and their users, in order to provide local computing capabilities on sensors, smart meters or other network-accessible devices. Following the same mentality where Cloud can be found in a distant location far away from the end user and Fog can be found closer to the end user, Edge Computing has also been associated with the term Mist Computing. As the name suggests, Mist Computing covers the computational and communication capacity available on the same level with the end devices. According to NIST [37], Mist Computing is defined as a lightweight and primitive type of Fog Computing which resides at the very edge of the network, bringing the layer of Fog Computing closer to the smart end devices. Mist Computing uses microcomputers and microcontrollers to feed into nodes of Fog Computing and theoretically into centralized (Cloud) Computing.

In light of this, Mobile Edge Computing (MEC) was developed as a key technology to assist wireless networks with Cloud Computing-like capabilities to provide low-latency and context-aware services directly from the network Edge [38–48]. Mobile Edge Computing, lately renamed as Multi-Access Edge Computing, was initiated under the umbrella of the European Telecommunication Standards Institute (ETSI) [47]. A key objective of the ETSI initiative is to standardize the APIs between the mobile Edge platform and the application service scenarios (augmented reality (AR), mixed reality (MR), intelligent video acceleration and Internet of Things gateway) and promote innovation in an open environment [38]. ETSI's reference architecture is largely based on the concept of Network Function Virtualization (NFV), where MEC applications can be offered as Virtualized Network Functions (VNFs).

2.1.5. Computing paradigms comparison

Almost all of the paradigms discussed above have as a common ground that they are offering remote computational and communication capabilities to the end devices. Furthermore, except from the Cloud, the rest of the paradigms are able to offer these capabilities at the edge of the network, as close to the end devices as possible. Nonetheless, there are some differences between them.

First of all, in terms of available resources, as we move farther from the end devices the available resources increase in quantity, with the Cloud having practically infinite capacity. Since a public Cloud may have dozens of data centers, each equipped with hundreds of servers, around the world, there is no actual problem of resource depletion. In contrast, in Fog, MCC, and MEC infrastructures, resource availability is mostly limited due to the fact that they are comprised of micro-data centers with few servers of lower capabilities than the ones that we usually find in the Cloud. On top of that, in edge infrastructures, we also find heterogeneous hardware resources with even lower resource availability such as wireless routers and gateways, street cabinets and Raspberry Pi's.

Secondly, delay can be another factor of comparison between the different infrastructures. As mentioned before, Cloud Computing is not always a feasible solution for providing low latency communication. To this end, the available infrastructure at the edge of the network is the most favorable option to reduce the communication delay. Nonetheless, since there are various levels of Edge at the WAN, LAN, or access network, different levels of delays can be produced, regarding where the computing resources are located. Obviously, going at the level of the access network, i.e., the extreme Edge or Mist, the delay is minimized since we do not have to account propagation and transmission delays involved in traversing the LAN, WAN or a backhaul network. However,

in this case, another factor rises; that of the energy consumption. The available hardware at the Mist is usually battery supplied, imposing the double burden of both limited resources and limited lifetime.

Thirdly, some of the paradigms were conceived under the scope of providing computational resources to specific applications. For example, Fog Computing was introduced to facilitate some of the top IoT application domains and vertical markets, such as energy, industry, transportation, agriculture, and healthcare [29]. On the other hand, MCC is mostly associated with providing remote computational resources to mobile applications, while MEC introduces the necessary flexibility to host multiple applications in the areas of video analytics, location services, IoT, augmented reality, optimized local content distribution and data caching among others [49]. Particular emphasis should be placed on the uniqueness of the MEC. In particular, even though MEC is oriented to cellular Radio Access Networks (RAN), it can be practically applied to any kind of access network. Furthermore, the way that MEC has been defined and standardized by ETSI, promotes an open environment where third-party developers, application and service providers can all participate together towards expediting the introduction of new applications targeting to respond to emerging user requirements.

Finally, the decision of which paradigm to follow, usually includes the requirements of security and confidentiality. Certainly, Cloud Computing as a popular and successful technology, has many safeguards and tools to provide a certain level of security and confidentiality. Nonetheless, several security threats still exist making Cloud-based security an active open-challenge. Additionally, sending data to the Cloud over the Internet can be susceptible to attacks. In contrast, by employing an Edge infrastructure, the necessary security and confidentiality can be attained since the data of the end devices usually stay within the local network.

From the above, the pros and cons of each computing paradigm can be extracted. Even though the MCC, Fog, and MEC can overcome certain limitations of the Cloud, they usually cannot be offered as a standalone solution. In other words, the notion of Edge Computing in general, did not emerge to replace the Cloud but rather to complement it. Thus, it is very important to create collaborative solutions (possibly utilizing more than two computing paradigms) that will enable a smooth continuum from the end device to the Cloud, with the goal to satisfy the stringent requirements of novel and future applications.

2.2. Use cases

Following the above definitions of the computing paradigms and the respective infrastructures, in this part of the survey we refer to some typical applications that leverage task offloading at the available resources at the edge of the network in order to increase their performance. These real-world applications can range from simple data processing to immersive multimedia applications. Following, we briefly describe the role of task offloading in the particular set of applications.

2.2.1. Immersive applications

Current developments in computer vision have made possible the launching of mixed reality applications, such as VR and AR, that can offer immersive experience even in wireless environments. At the same time, the development of increasingly advanced mobile devices such as smart glasses, can help us identify objects, superimpose contextual knowledge on our field of vision and create a three-dimensional view of the surrounding environment. As these devices become smarter, more and more sensor data in our environment can be aggregated, processed and served, requiring however high bandwidth and low response time communications. Hence, task offloading can be an advantageous solution for this type of applications.

Specifically, both Cloud and Edge-based task offloading mechanisms can be used in AR/VR applications [36,50–61]. The objectives include reducing the energy consumption in mobile devices, increasing the speed of computation intensive operations, reducing the average CPU load to overcome computation intensive tasks and improving the user's Quality of Experience (QoE).

2.2.2. Autonomous vehicles

Similar to the immersive media services, autonomous vehicles is another type of application that task offloading can be utilized. The key objectives here are to reduce the latency and the transmission cost and increase the efficiency of traffic management. Use case-oriented services concerning autonomous driving include: Highway Pilot, Parking Pilot, Fully Automated Vehicle and Vehicle on Demand [62].

Edge Computing is considered as the key technology in connected vehicles, adding computation capabilities and geo-distributed services to BSs and Edge devices distributed on the roadside. The idea is to analyze data from proximate vehicles and roadside sensors and broadcast messages to drivers at a very low latency [63]. For example, in an intelligent transportation system, low-level devices can be used for the decision-making processes of the transportation [64]. Specifically, the decision-making tasks can be distributed to Edge devices instead of sending all the data to a centralized server. Moreover, task offloading can enable real-time traffic management [65].

2.2.3. Robotics

Very complex robotic applications have been emerging during the last decade, related among others to autonomous mobile agents, manipulators and collaborative tasks. Efficient, safe and autonomous robot operation in manufacturing, health care, learning and exploration, requires running computation and memory intensive algorithms related to image processing, planning, localization, mapping and autonomous learning. Consequently, during the last few years, task offloading is gaining attention that has lead to the new paradigms of Cloud, Edge and Fog robotics [66–70].

Specifically, many offloading opportunities emerge in planning and SLAM (simultaneous localization and mapping) algorithms for robotic manipulators [71,72], mobile robots [73–77] and learning in general [78,79], among others. It is worth noting that there are already available commercial products that allow task offloading in robotic applications [80–82].

2.2.4. Video streaming

In general, the video streaming use cases fall under the content delivery network (CDN) [83] category. The key objective of CDN networks is to reduce the cost and the number of bits transmitted over the network, by maintaining an adequate QoE [84]. The mechanisms to reduce the overall cost and traffic while providing a high QoE in applications ranging from simple video streaming to HTTP, to Adaptive BitRate (ABR) and 360-degree video applications, can be further improved by applying task offloading techniques.

Offloading can be implemented on Cloud-based solutions, where appropriate resource allocation techniques can be used to increase user satisfaction [85] or deployment costs of the CDN networks [84]. Nonetheless, task offloading at the Edge can supplement the achieved performance. For example, multi-user mobile media delivery can be enhanced by enabling the gateways (i.e., BSs) to perform appropriate scheduling strategies [86]. An Edge infrastructure can also be used to facilitate the caching and transcoding mechanisms in a distributed fashion [87]. Regarding latency, data compression tasks can be offloaded at the Edge [88], removing the burden of local compression models and reducing at the same time the application response time [89]. Task offloading can be partially implemented by differentiating flows based on their quality and performing the video compression at the Edge, only for the high-quality video flows (e.g., 360-degree video streaming) [90].

2.2.5. IoT

The impact of task offloading can be maximized in the context of IoT applications. The reason is that the IoT devices are usually constrained in terms of available resources and battery capacity. Inevitably, only small and non resource demanding tasks can be executed locally. Task offloading in IoT usually focuses on reducing task execution time,

response time and energy consumption. IoT use cases that can benefit from task offloading can refer to health, agriculture, smart city, industry and energy related applications among others.

IoT, from the very beginning, has been largely based on Cloud-centric approaches in order to offload the tasks of data processing and analysis of massive data produced from millions of IoT devices [91]. However, the long delays added from the Cloud, combined with the introduction of new mission critical IoT applications, has pushed the academic and industrial community to take advantage of the Edge concept. Hence, local IoT Clouds have emerged, with the goal to maximize the number of offloaded tasks that can be executed in close proximity to the IoT devices [92] and to maximize the battery lifetime of the devices [93,94]. However, the scalability issues of the IoT market which is currently consisted of dozens of billions of devices often necessitates a Cloud–Edge collaboration during the task offloading [95–97].

2.2.6. Physical disaster management

In case of disaster management, the process of task offloading is crucial since it affects the efficiency of the rescue operations. In addition, the network can be unstable and simply offloading tasks to the Cloud could be difficult and require too much time. So, optimal offloading strategies to local services, rather than remote Clouds, would allow for precious time saving and preservation of battery of mobile phones, sensors and autonomous agents in the field.

Unmanned Aerial Vehicles (UAVs), which possess great mobility and versatility, are at the core of disaster management scenario by providing situational awareness and computing resources. But, as they are battery-powered, they cannot undertake the full computation of all the involved data and need to offload tasks to near Edge Computing servers. This challenge is addressed in different papers [98–100]. In general, task offloading in the context of disaster remains little explored [101,102].

3. Task offloading & challenges

In the previous sections, we have provided a short description of the task offloading, its infrastructural components, and the importance of this solution for new and emerging scenarios. In this part of the survey, a more detailed definition of task offloading is provided, while also, the typical objectives, the performance evaluation metrics, and the challenges encountered during task offloading are presented.

Generally, task offloading can be defined as the transfer of resource-intensive computational tasks to an external, resource-rich platform such as the ones used in Cloud, Edge or Fog Computing. Offloading the whole or part of the set of tasks to another processor or server, can be used to accelerate resource-intensive and latency-sensitive applications [65,90,103]. Task offloading is a complex process and can be affected by a number of different factors [24]. In particular, this process involves application partitioning, offloading decision making and distributed task execution [4,15,104]. A typical infrastructure involved in an offloading scenario is illustrated in Fig. 1. From this, it becomes evident that the network's Edge infrastructure creates an additional resource layer between the end devices and the external platform. This layer is capable of reducing bandwidth consumption on the backhaul, transport and Cloud networks, thus reducing any communication delays, supporting applications with real-time requirements, improving the energy efficiency and consequently increasing the lifetime of battery-powered devices. At this point, we should note that, for the rest of the paper, we refer to the term Edge Computing as the whole set of resources that can be found at the edge of the network, including the Fog and Edge nodes.

3.1. Granularity levels of task offloading

Task offloading aims at optimizing the offloading of computation intensive tasks from the end user device to a remote site, under various computational, communication and mobility constraints. The process of task offloading, as shown in Fig. 2, consists of (i) various hardware components, such as end user devices and Edge/Cloud devices, (ii) multiple computing processes, including task splitting and computational processing either locally or remotely and (iii) networking components for transferring data between the hardware components involved.

In more detail, as Fig. 2 illustrates, a mobile device can execute an application comprising of multiple tasks. The end device, through a task splitting process, decides which of these tasks should be executed locally and which ones should be offloaded to the Edge or Cloud infrastructures. This decision is based on a plethora of factors that are presented in the following sections, including the QoS requirements and battery lifetime of the device, among others. Following, the tasks that are to be executed remotely are transferred through the wireless access network to the gateway and from there to a remote physical machine (either at the Edge or Cloud), where they are executed following an appropriate computational approach (e.g., creating a VM or container). At the same time, the tasks that remain on the device are executed locally using the available computational resources of the end device. The last step is to combine the results of both local and remote executed tasks to provide the final output of the application. Based on this process, we describe the different types of task offloading according to the task splitting decision taken, i.e., the granularity level, as follows:

3.1.1. Partial offloading at the edge

In this type of offloading, part of the computing tasks is processed locally, while the rest is offloaded to the Edge. Partial offloading is typically the most effective, since it can benefit from both local and remote resources. Nonetheless, another level of complexity is added since it needs to be decided and scheduled which tasks should be offloaded while taking into account the possible energy and resource constraints of the end device.

3.1.2. Full offloading at the edge

In this case, all of the computing tasks are offloaded and processed at the Edge. Full offloading is usually translated into a simple resource allocation problem, where tasks can be executed on virtual machines or containers at the Edge. Energy-savings at the end devices can be maximized, however we need to take into account other sources of energy dissipation such as the transmission power of the device. Finally, a precise network path from the device to the Edge site, where the tasks are offloaded, has to be set up carefully, so as to comply with the possible QoE/QoS constraints.

3.1.3. Partial/full offloading at the edge and cloud

During this type of offloading, a collaboration between the Edge and Cloud infrastructures is established in order to execute the offloaded tasks. This type of collaboration can be proved advantageous in large-scale scenarios where the available Edge resources are not enough to host all of the tasks offloaded from the end users. Herein, the main challenge is the two-level task offloading decision. If a partial offloading mechanism is followed, the first level of decision lies on the local device, in order to decide the set of tasks that needs to be offloaded at a remote location. In other words, the local device has to decide which tasks can be executed locally in the device and which ones should be offloaded either at the Edge or the Cloud. On the tasks decided to be offloaded, a second-level of decision will be performed. In particular, the Edge will perform a second task partitioning, regardless of the type of offloading (i.e., partial or full) to determine the subset of tasks to be executed at the Edge and the subset of tasks to be executed at the Cloud. In the latter case, particular attention should be paid on the transport network that facilitates the interconnection of the two infrastructures and the delay constraints that it may impose.

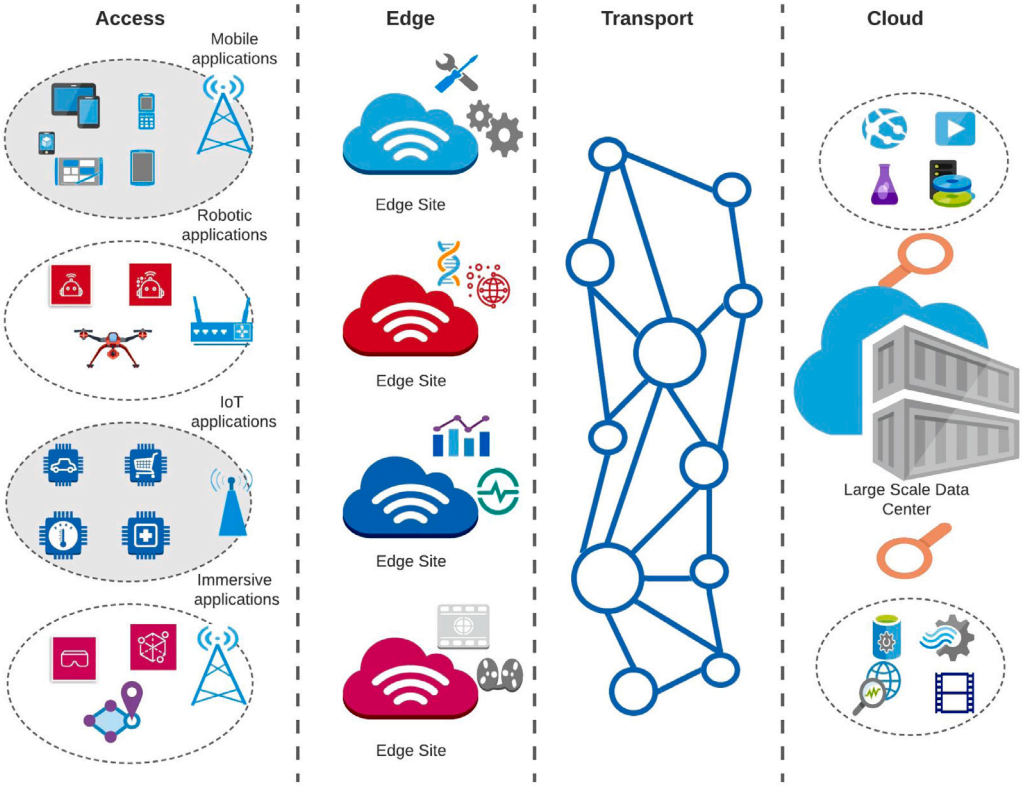


Fig. 1. Infrastructure components during task offloading.

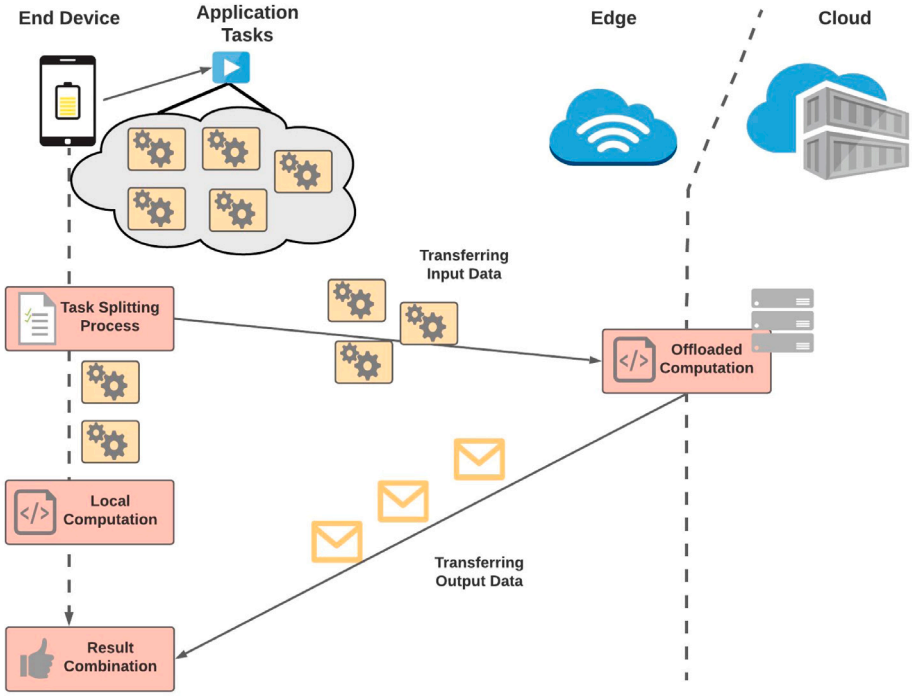


Fig. 2. Task offloading process.

3.2. Mobility of end devices

End device mobility is one of the most critical components when it comes to task offloading decision. End devices can either be considered as static or mobile for the time window which spans between initiating and finishing the offloading of their tasks. In the latter case, mobility adds another level of splitting decision, as it needs to be decided at which Edge site should the tasks be offloaded while the user is on the move (or not). Even though mobility is considered a challenge, it can generate a number of opportunities for the task offloading. First of all, it can initiate a load balancing technique to allow the system to provide the necessary services in distributed Edge

site scenarios [45,105,106]. Secondly, complementing mobility with appropriate prediction solutions can enhance the system's capacity, by finding the potential next associated BS/AP of the user [87,107,108]. This can be even more beneficial in a dense scenario, where the system can analyze the active users and their mobility patterns and allocate the resources in an online manner to existing and newly requested services. Moreover, mobility can benefit from handover mechanisms that can enable service migrations between BSs and their Edge servers. However, as the requirements of zero millisecond handover are studied by the 5G community, mobility with prediction mechanisms is starting to gain attention, in order to predict beforehand where the tasks should be offloaded. This behavior can be decisive for the overall performance in dense Edge deployments with multiple mobile users [109]. Based on the above, we firstly describe the different types of end devices according to their mobility level. Then, we present some mechanisms from the literature on how the Edge infrastructures adapt to the potential mobility of the offloading devices.

3.2.1. Static (low-mobility)

In this situation, the devices are considered static or relatively static (low levels of mobility) during the task offloading procedure. Apparently, this type of device mobility is considered trivial when it comes to studying the task offloading problem, as it does not impose any type of dynamicity to the network conditions. In certain cases, purely non-mobile devices like stationary IoT sensors are engaged in task offloading at the Edge, [110,111]. Other works apply this assumption on mobile devices, to reduce the complexity of their proposed offloading solutions; for example, the authors in [112] assume that the statistics of the utilized wireless links remain unchanged during the processing of the users' tasks, reflecting a relatively static or low-mobility scenario. In a similar manner, in [113] and [114], the time to transfer the task response from a cell to another is excluded from the total execution time of an offloaded mobile task, by assuming that the end device stays in the same cell during the task offloading process.

3.2.2. Mobile (high-mobility)

On the other hand, when the devices involved in task offloading are highly mobile, i.e., their movement has a direct effect on the network conditions considered for task offloading and the respective resource allocation, the problem becomes significantly more complex. Several studies consider mobility at the Edge [65,115–118]. Specifically, the focus is placed on the user contacts (inter-contacts) in which the user can offload the task, based on the mobility pattern and an opportunistic computing decision [119–123]. The opportunistic computing is taking advantage of the contact patterns regulated by the mobility of the devices (e.g., which Edge site the node is visiting and what type of interactions occur on daily basis), in order to determine the amount of computation to be offloaded to other devices. Opportunistic computing takes also advantage of the contact patterns regulated by the mobility of these devices, to determine the amount of computation to be offloaded to other devices. Furthermore, in a mobility scenario, users may either transfer their tasks to remote servers or peer devices, possibly through the gateways or even via the Edge servers [124–127]. For instance, mobility can influence the decision on which BS and Edge server to select and when to perform the handover [124]. When trying to minimize the execution delay, user mobility information needs to be combined with the task characteristics and resource availability, in order to make the best task scheduling decision [125]. This mobility information is usually captured by trajectory prediction models [128,129] that can actually uncover motion patterns of the users in real-time scenarios. However, in most of the studied scenarios [6,8,105,130], the tasks are usually offloaded to BSs that are in close proximity to the user's position and have sufficient resources to satisfy the time execution requirements. For example, when the tasks are lightweight and can be executed within a satisfactory time period, without migrating to a neighbor Edge site, then the execution of the task should be processed immediately and return to the mobile user. On the other hand, when the task requires significantly longer time, then the task could be split into sub-tasks and be transferred to neighboring BSs along the user's trajectory [6,8,131].

Most of the mobility tackling solutions integrate mechanisms to obtain the device's current and future positions, as well as tune the offloading infrastructure accordingly, to achieve the objectives described in the following subsections. Using the type of the mobility tackling mechanism as a criterion, mobility solutions can be categorized in the following classes:

Proactive — Behavior Related: Nowadays, services running on mobile devices, e.g., Google Location Services [132], constantly track and log the historic mobility behavior of the user; the proliferation of smartphone devices has made trajectory pattern crowdsourcing a reality. What is more, distributing intelligence at the Edge has allowed for logging the times an end user device connects and disconnects to a smart access point, thus extracting users' periodic movement patterns. Based on this data, mobility information can be estimated and leveraged towards predicting the users' position at any given moment [133,134]. Specifically, this mobility information can be extracted in a probabilistic way, by utilizing Mobility Markov Chains (MMC) to model the historic behavior of a user as a discrete stochastic process; in this way, the prediction of a user moving to a specific location depends on their previous visited locations and the probability distribution of the transitions between them [135]. Complementary to the regularity in the users' mobility patterns, a Markov model can be trained to estimate the expected network quality and the expected staying time under the coverage of each Edge server [136]. Then, one way to leverage the extracted information is in favor of bringing the Edge Computing and storage resources closer to the user; this can be achieved by proactively installing the services that the users will consume in the Edge servers located in the positions that they will most probably visit, thus reducing the network delay during task offloading [137–139]. The probability density function of the sojourn time, i.e., the time a user is expected to spend within the coverage area of an Edge site, can also be exploited towards predicting the user's next location and seamlessly migrating the service to be used for the task offloading appropriately [140]. For example, MAGA [141] is a mobility-aware mechanism for partial task offloading that falls in this category. Frequent mobility patterns are inferred by a tailor matching subsequence method and then a genetic algorithm is used for the offloading decision.

Proactive — Trajectory Related: Another way of proactively dealing with mobility at the Edge is by exploiting the user's ongoing movement characteristics, i.e., trajectory, duration and speed. By applying these characteristics on specific translational motion models, one can predict the location and the time of the next Edge server handover [142]. Apart from utilizing motion models, periodically receiving timestamped geolocation updates from a moving user, enables producing real-time travel information for route segments which can be used for trajectory estimation [143]. In a cooperative Edge infrastructure scenario, taking advantage of the mobility information can guide Edge servers to route the collected offloaded tasks to adjacent servers at the next location on the user's moving direction. In this way, when users arrive at the coverage area of the next Edge site, they receive the product of their completed offloaded task, with the minimum additional delay [144]. As an example, a two-step offloading mechanism for smart touristic services [145,146] is based on estimating the location and density of users. Every mobile device takes the initial offloading decision based on a dead reckoning technique and measurements of its WiFi signal strength. Secondly, at the Edge side, a Kalman filter is used to predict the number of users and a controller is responsible for the final offloading decision and the allocation of resources to VMs.

Reactive: The evolution of the Edge–Cloud continuum and the growing adoption it receives, has recently enabled network infrastructures to quickly and efficiently adapt to the rapidly changing user environment, in real time. When it comes to task offloading on the

move, Edge servers can utilize a central agent, located at the Cloud, to form a mobility-aware offloading infrastructure that tracks the users' position and optimally routes the task and its response through the closest server to the users' locations [147]. In a similar manner, the whole virtual server can migrate to the topologically closest Edge server to the user, reactively, every time a relocation is detected [148]. For instance, utilizing IP tracking, remote caching and the Software-Defined Networking (SDN) paradigm, can set the ground for efficient and timely task offloading as well; an SDN controller is able to track the user's network location, i.e., the Edge server in proximity and quickly react to changes in it by rerouting the offloaded task's response [149].

3.3. Task offloading objectives

When solving the task offloading problem, a number of different objectives may be applied, as the different stakeholders and actors (e.g., Cloud providers, Edge providers, Mobile Users and Service Providers) target a variety of goals. An objective function helps to formally and mathematically formulate these goals and guide the offloading solution. Objectives of the task offloading problem can be categorized as follows:

3.3.1. Delay

Task execution delay minimization is one of the main objectives during the task offloading problem [42,65,115,150–153]. Regardless of the type of task offloading, the overall goal is narrowed down to reducing the total task execution delay. This delay, can be broken down into a number of different delay contributors. The first source of delay is the task execution delay, coming from the task that can be either executed locally at the device, at the Edge or the Cloud. In the case of offloading the task at a remote Edge or Cloud site, we need to take into consideration the transmission and propagation delay at the various layers of the infrastructure (access, Edge, transport and Cloud networks), in both directions (i.e., sending the task and receiving the response). On top of that, processing and queuing delay at the various processing and forwarding devices should be taken into account. Finally, an additional delay contributor can be the time to optimally partition the task delay, during the task offloading decision [154]. The delay objective can be expressed as either the minimization of the average delay of each task [155] or the total delay of all the involved tasks of a mobile application. This objective is directly proportional to the available resources [115,150] and the network conditions [151,154].

The total execution delay can also be used to assess the impact of task offloading to the QoS achieved. Thus, according to the type of the application used and the part of the infrastructure under consideration, the task execution delay can be associated with: (i) the response time (i.e., the time duration from when a user requests a service until the service actually initiates [65,118,156]), (ii) the delay variation, in order to reveal how robust the task offloading solution can be, both over time and over dynamic traffic profiles, while also estimating the number of SLA (Service Level Agreement) violations noticed [107,157,158] and (iii) the network delay, including the four delay contributors (i.e., propagation, transmission, processing and queuing delay) at the different parts of the infrastructure [65,107,118,152,155,156,158–160].

3.3.2. Energy

The second most common objective during task offloading is the minimization of the energy consumption [161–165]. This energy consumption typically refers to the end devices [116,166–169]. The reason is that mobile and IoT devices are usually battery powered, thus a major concern is how to maximize the lifetime of the battery by reducing the device's energy consumption. Inevitably, it is reasonable to assume that, by following a full offloading approach, the maximum energy savings can be pursued. However, a number of other energy contributors need to be taken into account, even when a full offloading approach is followed. First of all, during the offloading, the transmission power, modulation and coding scheme, together with other radio parameters, needs to be taken into account, as they contribute in significant energy and consequently battery consumption [41,165]. This type of energy consumption can increase, especially when network conditions are not favorable. Secondly, by reviewing the full offloading from a complete, network-wide view, one can easily understand that the problem is simply pushed to the Edge and/or Cloud infrastructures. Thus, energy consumption minimization needs to be pursued at all layers of this end-to-end communication model [170,171].

To evaluate this objective, a number of different metrics can be used; the most common is the average power consumption measured by aggregating the power consumption on the hardware equipment used [159,172]. Alternatively, energy consumption can be used, expressed as the power consumption over time. Normally a minimization of power consumption leads to energy consumption minimization as well [107,173–175]. Furthermore, as the end devices are usually battery powered, the energy savings can be expressed as battery savings [170,176]. Finally, another way to provide the necessary environmental and economic sustainability is through minimizing the electricity cost [177]. Electricity cost depends on location and time. Hence, appropriate allocation of offloaded tasks potentially reduces the electricity cost, cutting down on operational costs while providing benefits to the environment [157,167].

3.3.3. Bandwidth/spectrum

The available bandwidth at the access network and how it can be shared by multiple users in order to offload the tasks, is also a significant constraint. However, due to the great influence it has on the task offloading performance, it can also be considered as an objective [166,176]. Due to the limited available bandwidth, especially in IoT networks and dense cellular networks, the careful allocation of spectrum becomes of utmost importance.

The objective of spectrum allocation is often associated with the transmission rate and power level of each end user [176], as well as the duration of the transmission of each device [151], in order to optimally share the available bandwidth. Thus, when trying to optimally deploy the available spectrum, an efficient metric is to evaluate the spectrum utilization in accordance with the number of offloaded tasks, power transmission and bandwidth consumption [64,152,173,174]. In view of the dynamic wireless conditions, the optimal scheduling of the bandwidth needs to follow the time-varying channel state and be also associated with the arrival rate of the tasks [166]. Throughput is another typical metric applied to evaluate the overall spectrum utilization, since it reveals how timely and efficiently the task can be offloaded at the remote infrastructures [170,178].

3.3.4. Load balancing

How to carefully allocate and schedule the available physical hardware resources is another objective to consider during task offloading [12,17,42]. Specifically, there is a high interest in path optimization, efficient resource usage and load balancing when solving the task offloading problem. The goal is to provide the necessary scalability by increasing the resource availability, increasing the number of offloaded tasks concurrently deployed at the Edge and/or Cloud [17,179], maximizing the resource sharing and fairness among multiple users [42] and facilitating the offloading of future tasks.

This objective can be translated into minimizing the overall resource usage (e.g., minimizing the average percentage of the computational and communication resource utilization), minimizing the maximum resource utilization of the infrastructure or minimizing the variance of resource utilization [95]. Load balancing can be applied either at each layer of the infrastructure or between the different layers. For example, the appropriate distribution of traffic between an Edge and a Cloud infrastructure, can be considered as an alternative load balancing objective [171,180].

3.3.5. Deployment cost

The task offloading problem can be often seen as a resource allocation problem where appropriate resources at the Edge and/or Cloud need to be reserved according to a deployment utility cost, in order to execute the offloaded tasks in a virtualized environment. This deployment cost can be modeled in various ways, each having a different interpretation. For example, the deployment cost can be defined as the aggregation of computational and communication resources that a set of tasks needs in order to be provisioned. This is typical when mobile or IoT applications needs to be complemented with specific network functions (e.g., security, compression, QoS) that is otherwise difficult for a user to achieve locally in his device [180–182].

In general, this cost can be expressed as the monetary cost for computing processing (e.g., $/CPU hour), memory (e.g., $/GB) and network bandwidth (e.g., $/GB/day), induced for using network resources. These costs can be associated with the Cloud and Edge providers and how and to whom they lease their infrastructure. Furthermore, since this cost usually follows a "pay-as-you-go" model, the number of physical resources used for the total number of tasks offloaded can reflect the deployment cost [152,157].

3.3.6. Model accuracy

All the objectives described so far, are some typical objectives that can be used regardless of the optimization solution/strategy followed. However, when it comes to Artificial Intelligence (AI) and Machine Learning (ML) -based approaches, an additional objective can be the model accuracy for the prediction of the behavior of the applications (e.g., mobility [183] and network related features [184]). Usually the AI/ML objectives are used as secondary or complementing objectives of the ones presented above [185]. In addition to that, regarding critical applications that build or use ML models during the task offloading, particular emphasis should be placed on the training time, inference time and the cost of the required computational power.

Model accuracy can be optimized by using the appropriate ML metrics. For example, regression metrics [186] can express how close the predictions of a model are compared to the actual values, by using the R-squared, Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) metrics. In clustering models, the evaluation metrics [187] measure the cohesion and separation of groups of observations. An example of such an approach is the Sum of Squared Error (SSE) metric, which aggregates the distance of each observation from its nearest cluster. When using classification ML techniques [188], typical metrics used to evaluate the accuracy are the precision (i.e., the percentage of relevant observations among the retrieved observations), Recall (i.e., the percentage of the total amount of relevant observations that were actually retrieved) and F-Measure (i.e., the harmonic mean of precision and recall).

3.3.7. Multi-objective

The typical objectives presented so far are usually conflicting, making task offloading a very challenging problem. For example, aiming to minimize the latency can lead to higher energy consumption on the end device, by deciding to execute the tasks locally and vice versa. When adopting load balancing objectives, offloaded tasks can be distributed among different Edge sites or between the Edge and the Cloud, in order to reduce the total delay and energy consumption. Similarly, when minimizing the deployment cost, offloaded tasks may be gathered in one single Edge site; this results in an uneven resource utilization that can also create significant congestion in the infrastructure and thus higher communication delays. Finally, when trying to optimize the spectrum allocation independently of the available communication and computational resources, it can result in poor offloading decisions in terms of delay, deployment cost and load balancing. Hence, multi-objective solutions can be used in order to explore the trade-off between the various objectives. The most common multi-objective approaches consider jointly minimizing the delay and energy consumption [117, 173,189–196]. Energy and latency optimization can also be combined with an optimal spectrum allocation through appropriate power and channel allocation [162,167,168,174]. Regarding load balancing, it can be jointly optimized with the delay, since both objectives are tightly correlated [42].

3.4. Challenges of task offloading

Achieving the aforementioned objectives of task offloading entails a series of challenges. In this section, we identify these challenges and classify them into two main categories.

3.4.1. Network dynamics challenges

Dynamic Network Conditions: The mobile and IoT networks are characterized as quickly varying access networks that create dynamic network and traffic conditions. This is a significant challenge that adds an extra level of complexity during the task offloading problem, since it is very difficult to pre-specify the behavior of the network. Aspects of noise, interference, fading and signal reflection can significantly impact the wireless communications, aggressively altering the overall throughput and delay of the wireless transmission. This necessitates an analysis and prediction of the network conditions, in order to accurately estimate when a task offloading decision positively affects the performance. Besides this, prediction can be combined with a resource allocation mechanism at the Edge and Cloud, since the amount of resources required for the task execution is directly proportional to the amount of traffic (i.e., the request rate) that will actually end up at the Edge or Cloud.

Dynamic User Behavior: Another level of impediment and dynamics, during task offloading, is added by the random behavior of the mobile users and how they employ their mobile applications. These behavioral aspects are very difficult to foresee and quantify, creating as result arbitrary user-based traffic profiles. A categorization of the mobile applications based on the users' preferences, the transmission patterns, the spatial and temporal correlation of the user generated traffic, as well as other traffic related characteristics, can be of utmost importance for the subsequent resource scheduling and allocation. Accordingly, machine learning and data analytic techniques should be applied to estimate the users' behavior and the rate of task generation.

Edge/Cloud Dynamics: Although the Edge and Cloud layers can work together in harmony, they still have their own dynamics. Cloud sites are centralized while Edge sites are distributed having only a local view of the network.This leads to different dynamics between these two layers. Specifically, contrary to Cloud, Edge has a spatial dynamic exactly because of its location awareness. Additionally, end devices can dynamically re-purpose and re-associate themselves to different applications by offloading different type of tasks or simply new devices could appear or disappear. This inevitably creates an additional dynamic factor for the Edge. Obviously, the initial task offloading and allocation decision over the Edge could be performed in an optimal way by the Cloud, since it possesses this centralized system view; however, the latter may not react timely to local dynamics. Therefore, Edge servers should meet the burden to locally decide to move services' tasks along time. Thus, a new challenge arises in order to (i) address these dynamics, (ii) create a consistent view of the tasks to be executed that Cloud and Edge should share, and (iii) place the different services at the right places in due time.

3.4.2. Resource allocation challenges

Task offloading is strongly affected by the resource allocation mechanisms that decide how and where the offloaded tasks will be executed in a remote platform. Thus, the task offloading and resource allocation decisions are coupled and should be addressed jointly. The main challenges this issue creates follow.

Partitioning Decision: The decision of which task to offload is the first and most significant challenge to address, as it comprises the core

Table 1
Comparison of Mathematical Optimization (MO), Artificial Intelligence (AI), and Control Theory (CT) approaches.

	MO	AI	CT
Stability			✓
Low complexity	✓		
Optimality	✓		✓
Online training		✓	
Reachability		✓	✓
Real-time decision	✓	✓	✓

of the task offloading problem. As shown in Fig. 2, when new tasks are generated by an application, an intelligent mechanism is required in order to decide whether the task should be executed locally or to be offloaded to a remote infrastructure. This partition decision of the tasks is associated with the task execution delay, the transmission delay and the energy consumption. A poor partitioning decision may result in performance bottlenecks regarding the execution of the application. Thus, a compromise between when and which tasks should be offloaded to the Cloud/Edge has to be explored, taking also into consideration any possible transmission costs in terms of energy, delay and money.

Resource Availability: The performance of an application is closely dependent on the resources available at the end user/Cloud continuum. In general, as we move towards the core of the infrastructure, the available resources increase in amount, paying the price, however, of a higher application delay. Hence, sharing these resources is a crucial challenge which needs an efficient resource allocation and management mechanism that will be able to guarantee the performance requirements. Thus, alongside the offloading decision, the resource allocation mechanism should fulfill various functional and non-functional requirements. The primary goal of resource scheduling is the respect of the QoS requirements of the application. Additionally, the resource allocation should guarantee important properties, such as stability, reachability, safety and robustness against internal uncertainties and external disturbances. In terms of functional requirements, the resource allocation strategy should be implemented with commercial or open-source resource orchestration tools, that enable scalability, interoperability and the transparent development of the applications over heterogeneous hardware and software technologies.

Performance Modeling: Measuring the performance of a task offloading solution is an additional challenge. The task offloading problem can be modeled as a system where the energy and/or delay are the typical output variables and the available computing resources (e.g., CPU, memory), incoming requests and network bandwidth are the input variables. In most of the current studies, the proposed performance models are single-input/single-output, empirically derived or fixed. Although this assumption is realistic, the processing time of an offloaded task depends on many time-varying parameters, which are usually not easily measured. On the other hand, multi-input/multi-output models are more accurate, but the identification process is usually strenuous. Specifically, the offloading decision performs adequately only for specific operating conditions, being unable to guarantee stability under fluctuating workloads and heterogeneous communication infrastructures, such as in IoT. Hence, this system model should be adapted in order to include the performance metrics, expressed as state variables, that can be regulated by the control parameters (i.e., the input variables). This framework will be capable of capturing structural changes interpreted as discrete jumps in the dynamics, e.g., user mobility, changes in network conditions and addition/removal of Edge servers.

Task Management: As stated before, offloading tasks to the Cloud follows a centralized approach in which the Cloud infrastructure serves the whole set of tasks coming from the access network layer. In contrast, at the Edge layer, the infrastructure is usually distributed in multiple geographically dispersed Edge sites. Obviously, this is one of the core advantages of Edge Computing, as it creates a local efficiency by executing the tasks and effectuating actuation in minimal time. However, at the same time, a meticulous design of the task management control modules is required at the Edge. The placement of controllers and their mapping to the sites that they will serve, the decision of which task is going to be offloaded at each site and how the load is going to be distributed between the sites, are some of the challenges that fall in this category.

4. A taxonomy of task offloading approaches

This section lists the most prominent task offloading solutions that have been proposed in the literature. These algorithmic solutions can be divided into three categories: (i) Mathematical Optimization algorithms; (ii) Artificial Intelligence techniques and (iii) Control Theory-based approaches.

4.1. Methodology comparison

Before providing the most common approaches for each category, it is worth investigating the advantages and disadvantages of each category and the level of their efficiency. Accordingly, we present the main characteristics of each category, while Table 1 summarizes the main features supported.

Mathematical optimization is the most common solution category applied in resource allocation and scheduling networking problems. The reason lies in the fact that, traditionally, these types of problems can be mathematically formulated and solved, using a great variety of existing solutions. Usually, the main goal for this category of algorithms is to find the optimal solution among a set of possible solutions. For example, in the context of task offloading the mathematical optimization approaches will have to appropriately model the input (e.g., Edge/Cloud infrastructure, end users, available resources, task distribution size and duration) and, according to a certain objective, decide when and where the tasks should be optimally offloaded.

The optimality can be achieved through exhaustive search optimization solvers in the expense of a high complexity and execution time. Nonetheless, mathematical optimization approaches can reduce their time complexity when relaxing any hard constraints of the input and altering their final goal into finding a sub-optimal but fast and real-time solution. Even though these types of solutions can fit and be used in real scenarios, they sometimes suffer from their static nature and inability to adapt and model the dynamic challenges inherited from the problem at hand. Under such circumstances, the algorithms should be re-executed and re-customized every time a change happens in time and/or space (e.g., dynamical arrivals of end users, mobility and equipment failures).

AI task offloading mechanisms have also seen great progress nowadays. Data driven models, learning from batch or online data, provide real-time task offloading decisions and elastic resource allocation. Decisions are made by generalizing historical data, recognizing in an automatic way the prevailing data patterns and evaluating the possible destination states of the main actors in the Cloud/Edge environment. The state of Edge infrastructures includes the status of computing nodes in terms of resource utilization, the number of application requests and the user requirements contracted as SLAs.

Contemporary AI and ML task offloading is characterized by flexible adaptation and automatic learning. ML models have the advantages that they are not explicitly designed by human experts and they are self-trained based on the available data. In addition, they can handle multi-dimensional and multi-variety data in a unified way and they are capable of identifying hidden patterns. The main weakness of AI-based models lies in the case of significant inconsistency between training and testing data properties, which may lead to performance degradation. This means that the data should be selected and gathered with diligent attention to detail and special emphasis should be given to the data

preparation tasks of synchronization, transformation, and normalization. Lastly, we should take into consideration that the large amount and high frequency of data make the training model a computational heavy and resource-intensive process.

System Theory provides various models for describing the operation of a process. Additionally, Control Theory provides many formal methodologies to analyze and control the performance of the process. Both of them have been widely used for industrial processes while, during the last decades, they have been introduced to computer networks. System theory provides black- or gray-box training algorithms, namely system identification, to compute multiple-input-multiple-output (MIMO) models that capture the system dynamics of continuous or discrete systems. However, system identification must be performed offline and the computed model may have low accuracy. The control-based task offloading solutions enable real-time decisions against the dynamic network and workload conditions. Additionally, apart from reaching an optimal operating point, the control-based offloading solutions can guarantee various system properties, such as stability and reachability. The guarantee of stability means that the system will reach specific operating conditions and will remain on them against any disturbance. The reachability property means that, given the current system state, we can compute all possible destination states. Although, the complexity of a controller increases with the complexity of the system model (linear or not) and the properties to be guaranteed, the design is an offline process and the real-time application of control law is simple.

4.2. Mathematical optimization algorithms

The task offloading problem is usually formulated as a mathematical problem, which tries to find an optimal or near-optimal solution. The problem can be formulated by defining the objective function as described in Section 3.3 and the optimization strategy used. These strategies may include Mixed Integer Programming, heuristics, meta-heuristics and game theory approaches, among others. Following, we present the main optimization strategies found in the literature, while a summary of them along with the objective of the study, algorithm developed and type of offloading is listed in Table 2. Furthermore, Fig. 3 illustrates the key components of the existing mathematical optimization approaches, shedding light into the well and less explored proposed solutions.

4.2.1. Mixed integer programming

Mixed Integer Programming (MIP) formulations provide a flexible and mathematically precise way of formulating many real-world problems. Specifically, integer programming is a commonly used technique for resource allocation and scheduling in wired and wireless networks. The two main problem types that MIP addresses are: (i) network synthesis and (ii) resource assignment problems [208]. MIP optimization approaches facilitate also the introduction of a multi-objective function optimizing more than one goals under various offloading constraints (e.g., delay, energy and load balancing). Hence it can be often used as an optimization strategy during the task offloading problem. Usually, MIP provides a linear objective function (MILP), where at least one of the variables takes integer/binary values. Even though these types of problems can provide the optimal solution, they can be very complex or even computationally intractable for large scale experimentation scenarios. However, they can often be used as benchmark approaches during the performance evaluation. For example, in the context of task offloading, they can be used to minimize the weighted amount of mobile energy consumption in a multi-user system, under latency constraints [192]. Regarding delay, the objective function of the MIP can include both the transmission and processing delay, especially for IoT mission-critical applications in an Edge–Cloud collaboration [197].

In case the objective is non-linear, a Mixed Integer non-linear (MINLP) or quadratic (MIQP) programming formulation is modeled. Similarly, these types of problems allow the formation of multi-objective functions. For instance, a system cost representing the weighted cost of delay and energy consumption among all available users, can be expressed as a non-linear objective function in a mixed Cloud–Edge task offloading environment [198]. In case only the energy exists in the objective functions, typically the latency requirements can be imposed as constraints along with other various conditions (e.g., power consumption levels, channel states and resource heterogeneity) [199].

4.2.2. Heuristics

Heuristic approaches can introduce fast but sub-optimal solutions. The main advantage of heuristics is that they are simple algorithms devised to address the problem at hand, with low execution time. In contrast with MIP algorithms, they do not require specialized optimization tools to be solved and can rather be expressed as pseudo-code, easily implementable in any programming language. To this end, heuristic solutions are very popular to be applied in the task offloading problem. These types of solutions can range from optimizing the offloading decision of the user while minimizing the overall cost of energy, computation and delay, by applying appropriate relaxation and randomization techniques [194], to optimizing the resource allocation at the Edge, by considering a Cross-Entropy optimization approach [162]. Heuristics can also be used to optimize non quantifiable parameters, such as QoE in a Cloud–Edge collaboration [197], by satisfying the various computational and bandwidth restrictions, applying appropriate fairness and popularity techniques [85].

The efficiency of the heuristic becomes much more evident when the task offloading problem is modeled as a non-linear constrained optimization problem, or when the experimentation covers large-scale offloading scenarios [162]. In this case, greedy heuristics can be used to estimate the exact solution [42,85,200].

4.2.3. Game theory

Lately, there has been also a use of game theoretic approaches to deal with resource allocation problems. Through game theory, the task offloading problem can be introduced as a resource allocation game. For example, the problem of the partial task offloading in a multi-user, Edge Computing infrastructure and a multi-channel wireless interference environment, can be formulated as an offloading game [201]. This game tries to maximize the spectrum efficiency during offloading by allocating the proper channel to each user/player. The specific approach can be complemented with a second matching game that will aim to maximize the efficiency of resource allocation at the Edge, by appropriately selecting the right Edge servers [196]. A multi-step/slot game theoretic approach can be followed in order to find the optimal state, expressed as the Nash Equilibrium. Specifically, in each step the end user/player can make a decision on whether to offload their tasks in order to reach a potentially optimal offloading. A similar slotted approach can be followed by treating each user as a player with the goal to optimize the CPU-cycle frequency and offloading decision, in order to maximize the energy efficiency. Game theory has also been used in an Edge–Cloud interplay, where players can be considered as the corresponding infrastructures [202]. In particular, by formulating a Stackelberg game, a leader player is assigned the goal to maximize a utility function expressed in order to obtain the optimal revenue for the Edge and Cloud providers, while satisfying the delay requirements. A different objective can be considered in an Edge–Cloud collaboration, where the two infrastructures comprise the players of the problem and try to minimize the overall energy consumption under delay constraints [203], by using a potential game [209].

Table 2
Taxonomy of mathematical optimization task offloading algorithms.

Reference	Optimization objective					Algorithms					Mobility		Offloading		
	Delay	Energy	Bandwidth/ spectrum	Load balancing	Deployment cost	MIP	Heuristic	Game theory	Contract theory	Local search	Static	Mobile	Partial	Full	Edge–Cloud
[41]	✓	✓					✓				✓			✓	
[42]			✓				✓	✓			✓		✓		
[65]	✓			✓		✓						✓		✓	
[85]			✓		✓	✓	✓				✓				✓
[89]	✓	✓					✓				✓		✓		
[116]		✓					✓					✓		✓	
[117]	✓	✓	✓	✓						✓		✓	✓		
[121]	✓	✓	✓						✓		✓			✓	
[150]	✓		✓	✓		✓	✓			✓	✓		✓		
[151]	✓	✓								✓	✓			✓	
[152]	✓		✓	✓	✓		✓				✓				✓
[154]	✓		✓	✓			✓				✓		✓		
[155]	✓	✓								✓	✓			✓	
[161]	✓	✓					✓				✓		✓		
[162]	✓	✓	✓						✓	✓	✓		✓		
[163]	✓	✓					✓				✓		✓		
[164]		✓								✓	✓		✓		
[165]	✓	✓		✓		✓				✓	✓		✓		
[166]	✓	✓					✓				✓			✓	
[167]	✓	✓					✓				✓			✓	
[168]	✓	✓		✓			✓				✓			✓	
[169]	✓	✓	✓	✓			✓				✓			✓	
[170]	✓	✓								✓	✓			✓	
[173]	✓	✓				✓				✓	✓				✓
[174]	✓	✓				✓				✓	✓				✓
[176]		✓	✓				✓				✓		✓		
[189]	✓	✓								✓	✓		✓		
[191]	✓	✓	✓				✓				✓		✓		
[192]		✓				✓				✓	✓		✓		
[193]	✓	✓	✓		✓	✓				✓	✓			✓	
[194]	✓	✓					✓				✓			✓	
[195]	✓	✓				✓					✓			✓	
[196]	✓	✓				✓		✓			✓		✓		
[197]	✓	✓				✓	✓				✓		✓		
[198]	✓	✓			✓	✓					✓				✓
[199]	✓	✓				✓					✓				✓
[200]	✓	✓					✓				✓				✓
[201]	✓							✓			✓			✓	
[202]	✓							✓			✓				✓
[203]	✓	✓						✓			✓				✓
[204]	✓	✓							✓		✓		✓		
[205]			✓	✓					✓		✓				✓
[206]		✓	✓		✓				✓			✓			✓
[207]	✓				✓				✓			✓			✓

4.2.4. Contract theory

Naturally, task offloading introduces conflicts between the participating parties; for example, on the one hand, users and devices seek to maximize energy and spectrum efficiency while on the other, small cells and Edge servers try to minimize consumption of their own resources, like battery capacity and computing power. Conflicts like these might cause reluctance by third parties to participate, which could subsequently raise barriers to the development of attractive traffic offloading solutions. Contract theory is an approach originating from real world economics, that dictates the design of contracts to achieve cooperation between the conflicting sides. In a broad sense, contract theory studies the design of formal and informal agreements that motivate agents with conflicting interests to take mutually beneficial actions. In the wireless networks domain, the agents include the BS, service provider and the spectrum owner, as well as the small cells, smart devices and users [210]. The late boom of contract theory applications in task offloading has managed to deal with many early challenges of the field; for instance, combining contract theory with game theory and a monetary rewards system, can eliminate the influence of information asymmetry in a user–Edge server relationship [204]. Similar incentives can be utilized when dealing with small-cell base stations (SBSs) and heterogeneous ultra-dense networks (HetUDNs) [205]. When the goal is to optimize bandwidth allocation in data offloading, dynamic programming concepts can integrate with contract formulation, as in the UAV to macro base station (MBS) offloading scenario, described in [206]. In the case of opportunistically offloading part of the cellular traffic to coexisting networks, towards alleviating the overload problems caused by traffic demands, a contract theory-based incentive mechanism can motivate users to leverage their delay tolerance in exchange for service cost [207].

4.2.5. Local search

Local search algorithms adopt mechanisms of perturbation to explore neighbor solutions in the search space, that allow to gradually converge to local optimum solutions. Due to the problem-agnostic nature of the local search algorithms, they can be used as a component of heuristics and meta-heuristics in order to provide solutions very close to optimality [211]. For instance, a one-dimensional local search algorithm can be used for a partial task offloading solution, with the goal to minimize the average execution delay, expressed as a Markov chain process, under the energy constraints imposed by the device [155]. When both energy and latency constitute the objective of the partial task offloading problem, a univariate search technique can be used [189]. This type of search allows to transform a non-convex problem into a convex one, by finding a local optimum solution. An iterative local search can further reduce the gap with an optimal solution, where multiple iterations of the local search can result in

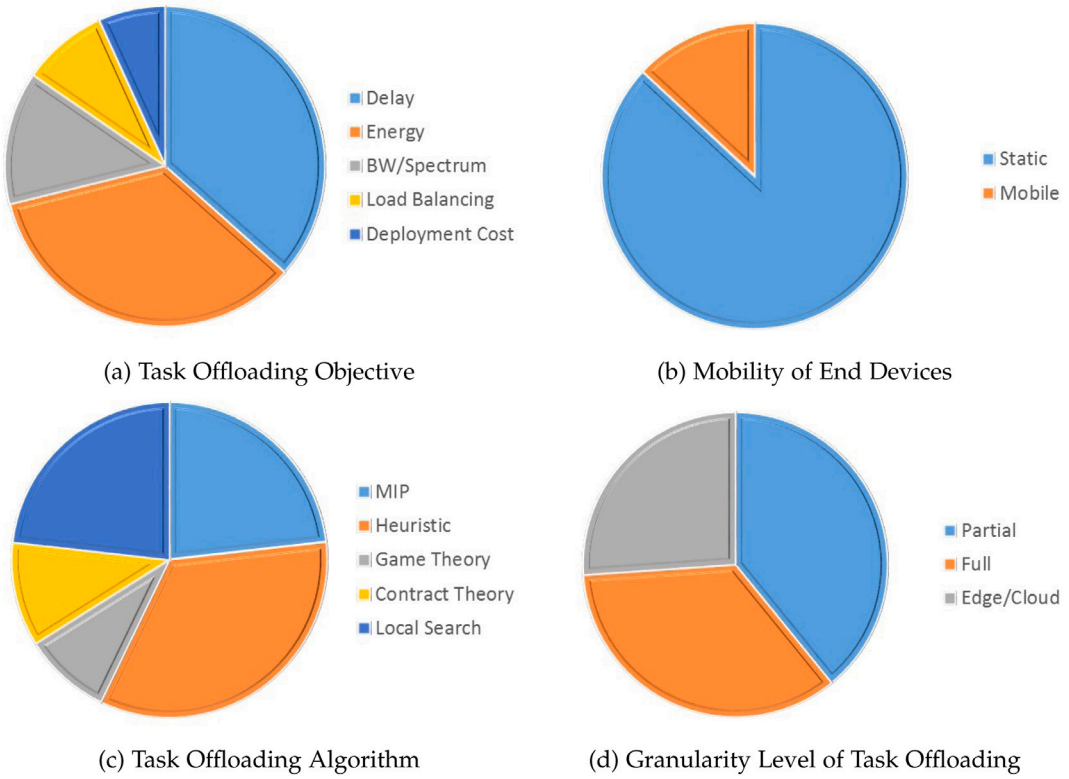


Fig. 3. Summary of mathematical optimization task offloading algorithms.

a better resource allocation of computing and channel resources in a partial offloading scenario [174]. When the goal is to optimize the energy sustainability of the end-user devices, by selecting the proper Edge resources and at the same time to minimize the execution time of the allocation, a simple bi-section search algorithm can be used [151].

4.3. AI-based optimization algorithms

In the above section, we provided traditional mathematical and algorithmic approaches to derive the optimal or near-optimal solution, in the context of task offloading. However, these approaches may suffer from the following issues: (i) Most of the solutions investigated so far fail to take into consideration the dynamic network conditions. Since this is a random variable, it is difficult to estimate and reflect this behavior during the allocation of the Edge and Cloud resources and during the task partitioning decision; (ii) The traditional approaches are rather opportunistic, addressing the challenges of the task offloading in a short-term scale. However, in this manner, we cannot capture the long-term time and space varying conditions in all the layers of this end-to-end communication model. In other words, the solutions presented above lack the “intelligence" to better adapt holistically to the inherent challenges of the problem at hand. This prepares the ground for using artificial intelligence techniques for the task offloading problem.

Artificial Intelligence (AI) techniques include multi-disciplinary techniques from machine learning, consensus-based and constraint-based algorithms and they have been widely used in different computer systems and network scenarios [8,185,200,212,213]. AI techniques are becoming successful alternatives also for solving optimization problems that include the mathematical formulation of uncertain, stochastic and dynamic information, thus making them excellent candidates for the task offloading problem. Furthermore, AI can potentially reduce the complexity by enabling recursive feedback-based learning and local interactions and thus faster speed in seeking sub-optimal solutions than traditional techniques [8,185,212]. For example, during the task offloading problem, by learning from data and tasks distributed across the Edge infrastructure, AI can enable a smart, real-time, and dynamic resource management framework [11,214–216]. On another perspective, AI techniques can also be applied to avoid costly data offloading, by enabling data estimation or prediction, like in dual prediction approaches [217].

Similar to traditional mathematical optimization solutions, when using AI, a problem can be formulated by defining the objective function and the algorithmic strategy to be followed. Following, we present the major AI techniques used in the literature to address the task offloading problem. A summary of the related work with the objective of the study, the algorithm developed and type of offloading, is listed in Table 3, while Fig. 4 illustrates the distribution of the key components of the existing AI optimization approaches.

4.3.1. Machine learning

As a sub-category of AI, Machine Learning (ML) gives devices or computer systems the capability to learn useful patterns and behaviors from historic data and make decisions about new ones. The models are built without explicit programming; in the case of ML parameterized models, such as Linear Discriminant Analysis, Logistic Regression and Naive Bayes, the models are built by tuning a fixed number of parameters of a predefined mapping function. In the case of non-parametric models, such as the RBF-kernel Support Vector Machines, Decision Trees and K-Nearest Neighbor, the models use a flexible number of parameters with no prior knowledge about the data distribution and mapping function. In both cases, a mapping function is a function that maps the independent data variables to the dependent variables, i.e., the variables the model predicts.

ML models can be divided in supervised, unsupervised and reinforcement learning ones, based on the available training data. A prominent ML subfield is Deep Learning, which involves Artificial Neural Networks (ANN) with multiple layers of representation. ML models have been used successfully to overcome the challenges of task offloading and resource allocation, as described below.

Table 3
Taxonomy of AI-based task offloading algorithms.

Reference	Optimization objective						Algorithms			Mobility		Granularity		
	Delay	Energy	Bandwidth/ spectrum	Load balancing	Deployment cost	Model accuracy	Machine learning	Population	Constraint	Static	Mobile	Partial	Full	Edge–Cloud
[218]	✓	✓				✓	✓			✓		✓		
[219]	✓	✓					✓				✓			✓
[220]	✓	✓				✓	✓				✓			✓
[221]	✓	✓				✓	✓			✓				✓
[222]	✓	✓			✓	✓	✓			✓		✓		
[223]	✓					✓	✓			✓				✓
[224]	✓						✓			✓			✓	
[225]	✓						✓				✓		✓	
[226]		✓					✓			✓				✓
[227]	✓	✓					✓			✓				✓
[228]					✓		✓			✓				✓
[229]		✓					✓			✓				✓
[230]	✓	✓		✓			✓			✓			✓	
[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]	✓	✓						✓		✓				✓
[258]	✓								✓		✓		✓	
[259]	✓	✓		✓		✓			✓	✓				✓
[260]	✓	✓	✓						✓	✓				✓
[261]	✓				✓				✓	✓				✓
[262]	✓	✓			✓				✓		✓			✓

Supervised ML Models: Supervised ML models include classification and regression models. In classification, the model predicts classes while in regression the model estimates continuous values. The offloading decision can be formulated with a multiclass classification method and the resource allocation with a regression model [218]. Classification and Regression Trees (CART) [219] have been used to select the fittest Edge device for offloading, while minimizing time and energy by taking into consideration parameters such as the authentication, confidentiality, integrity, availability, capacity, speed and cost. Classifiers such as JRIP and J48 [220] have been used for context-sensitive offloading in a Mobile Cloud Computing environment using a robust profiling system. Logistic regression [221,222] has been used to calculate the load of each Edge node and enhance a dynamic resource allocation strategy. A different classification approach classifies Edge applications into classes of services [223], based on their QoS requirements, and maps them to Edge and Cloud resources. The Apriori algorithm [224] has also been used to generate rules for every task, in order to select the Edge node that offers the minimum completion time.

Linear Regression [225] has been used to predict the total processing duration of each task on each candidate Edge node, in order to offload entire tasks to one Edge node instead of a local execution. Linear Regression [226] has also been used to predict over-loaded and under-loaded nodes, in order to facilitate a live migration process of tasks. Gaussian Process Regression [227] has been proposed to predict the future workload of the tasks, allowing the deployment of new, delay sensitive applications and reducing energy consumption, blocking of requests and latency. The dynamic characteristics of applications and the complex Edge/Cloud Computing environment, have been modeled with the Support Vector Regressor [228] and the K-Nearest Neighbor Regressor [229] for future load prediction and energy efficient utilization of Edge servers respectively.

Unsupervised ML Models: Clustering models discover groups of objects that are similar, close and dense or share some common properties. Clustering models are differentiated from Classification models in that they do not require annotated data for training. Regarding task offloading, clustering approaches have been used to group resources based on the distance between Edge nodes, wireless and computational resources [230,231] in order to minimize the response delay. In the same fashion, Edge sites can be grouped for different task resource demands [232] and Edge servers can be grouped using an analysis of the allocated computing resources [233]. Unmanned aerial vehicles are also clustered to enable efficient multi-modal and multi-task offloading [234] and IoT users according to their priorities [235]. The

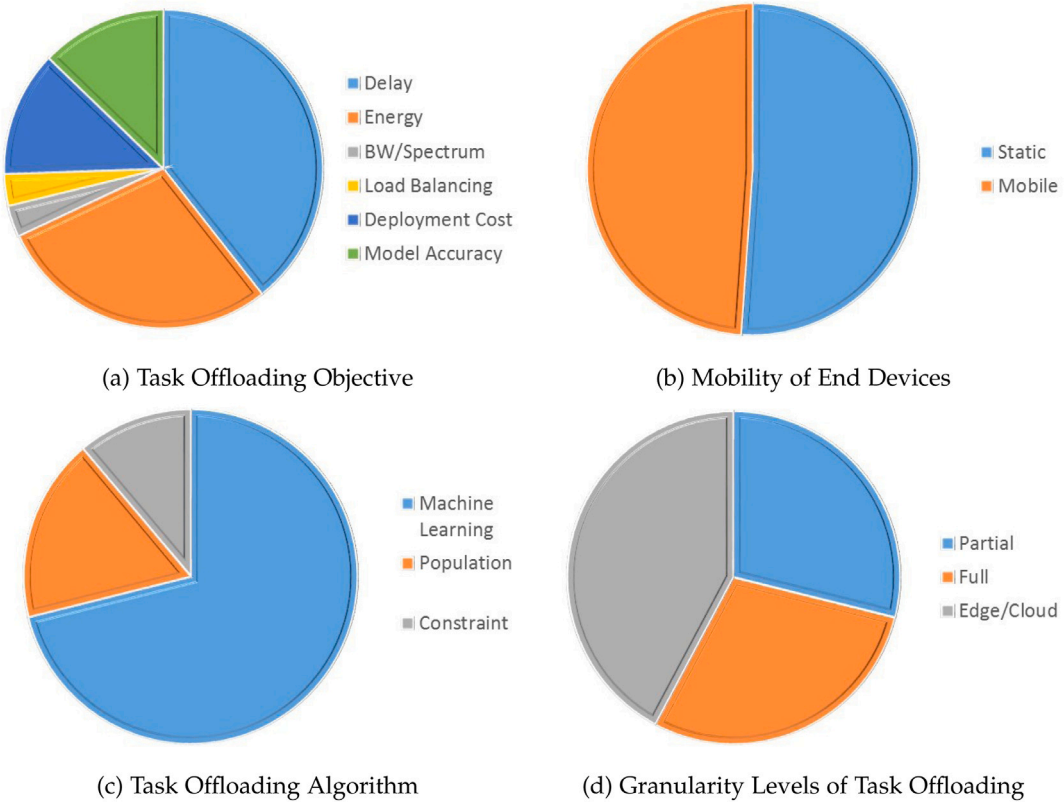


Fig. 4. Summary of AI-based task offloading algorithms.

dependencies between tasks can be represented by a graph and, by following a fuzzy clustering [236], makespan (i.e., the time difference between the start and finish of tasks), monetary and energy costs can be minimized. The K-means clustering method [237] can provide efficient task scheduling, thus increasing the utilization of the Edge devices, based on the type of resource requirements in terms of CPU, I/O and communication. Lastly, a policy-based clustering approach [238] can provide energy efficient task offloading solutions, by organizing the interactions among the Edge nodes.

Deep Learning: Deep Learning has gained popularity in multiple decision and scheduling problems because of highly accurate outcomes, especially when large amounts of data are available. In Edge and Cloud Computing, large amounts of data are being collected by resource monitoring tools, application logging mechanisms and network sniffers [263]. A modular deep learning model can integrate different sources of data, manipulate the data observations with hierarchical layers of representations and extract generalized knowledge that goes beyond the historical observations.

Deep Learning can provide timely and accurate task offloading decisions, based on the resource usage of processing Edge nodes, the workload and the QoS constraints defined in SLA [264]. A deep learning model works as a function approximator that takes as input the current infrastructure and workload status and outputs the appropriate processing nodes where each tasks will be offloaded. Further outputs of the deep learning models include the decisions for vertical or horizontal scaling up and VM migration, in order to guarantee the smooth operation of tasks execution in a dynamic and quick-change computing environment. Specifically, Deep Learning models have been implemented to minimize the computation and task offloading overhead in varying network conditions and limited computation resources [239]. A Deep Learning model that also addresses the challenges of speed, power and security, while satisfying the quality of services, has been proposed in order to determine the combination of different devices and dynamic tasks [240]. In addition, close to the optimal joint offloading decisions, bandwidth allocation can be generated with a distributed deep learning-based offloading algorithm for Edge networks [241]. Furthermore, the challenge of energy efficient task offloading with Deep Learning has been studied in the context of the internet of vehicles [242], users' equipment [243] and specifically for delay-sensitive and computation intensive tasks in Edge Computing networks [244].

Deep Reinforcement Learning: Generally, reinforcement Learning models take actions in an environment in order to maximize a cumulative reward or minimize expected loss. Reinforcement Learning is usually combined with Deep Learning in order to generalize with previously unseen data in terms of environment, states and actions. In the Edge Computing context, a Deep Reinforcement Learning model has been implemented to make the binary offloading decision on whether the offloading will take place partially locally or fully remotely to an Edge server [245]. Deep Q-Networks [246,247] have been proposed to automatically infer the offloading decisions, in order to optimize the system performance. Furthermore, Deep Q-Networks have been enhanced to capture the sequence of data with long short-term memory layers [248], for mobile tasks in a large-scale heterogeneous Edge environment and Gated Recurrent Unit layers [249] in multi-Edge networks.

4.3.2. Population-based methods

Population-Based methods include a wide range of nature-inspired algorithms and provide close to optimal solutions in combinatorial problems, following a metaheuristic approach. Two main subcategories of Population-Based methods are the Swarm Intelligence and the Evolutionary Algorithms. Both of them have been proposed in order to provide efficient solutions in task offloading challenges.

Swarm Intelligence (Consensus-based): The Swarm Intelligence properties make it a practical design model for algorithms that solve increasingly complex problems. In general, swarm algorithms strive to allow the entire system to converge into a global consensus state, while retaining the ability to perform assigned individual tasks in the

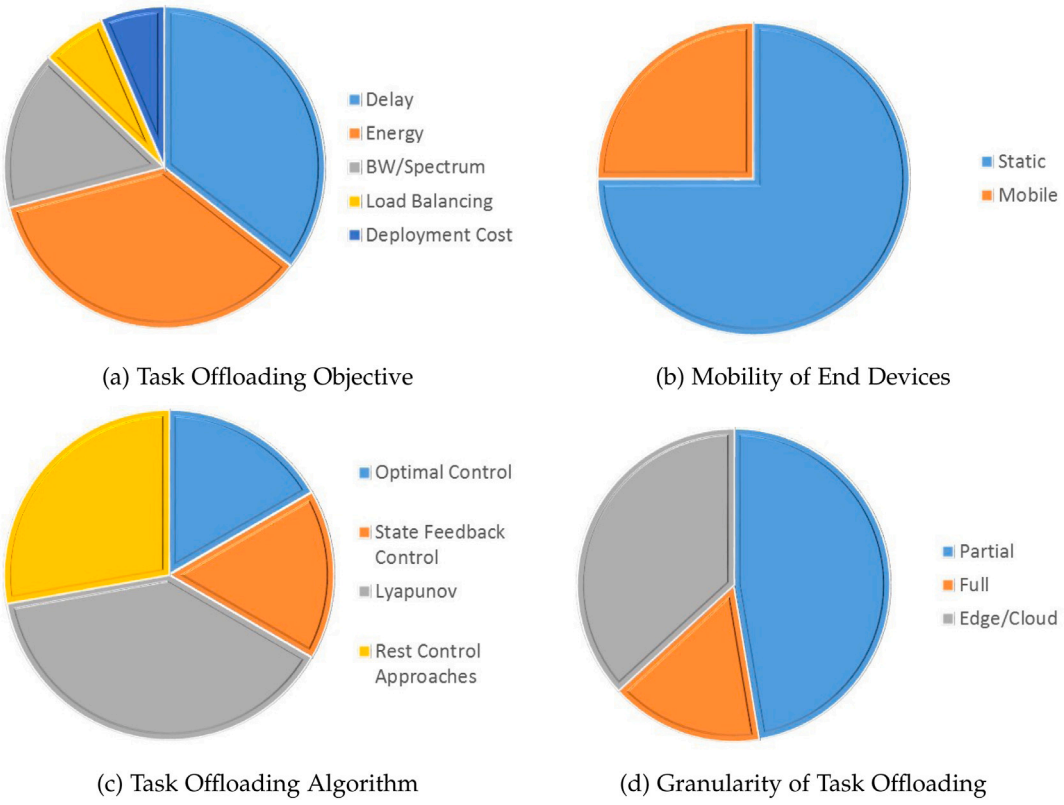


Fig. 5. Summary of control theory-based task offloading algorithms.

swarm. Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) are the most common Swarm Intelligence algorithms. ACO can be applied for efficient task scheduling due to its strong global search ability [250] and improve the response time of IoT applications by distributing effectively the tasks over the Edge nodes [251]. On the other hand, PSO can be used to minimize both the transmission and the processing delay, as a means to minimize the total end-to-end delay during a partial task offloading at the Edge [252]. In this case, PSO can also incorporate other task offloading key mechanisms, such as VM migration and transmission power management, to minimize service delay as efficiently as possible, to provide high QoS for different application profiles and to remain computationally feasible. Last but not least, PSO has been used to jointly minimize energy consumption and completion time for high-quality solutions [253,254].

Evolutionary Algorithms: Evolutionary Algorithms are based on natural selection principles, such as reproduction, mutation, recombination and selection. They perform a lot of iterations on a set of candidate solutions, aiming for the closer to optimal solutions to survive as much as possible, while the unfit solutions tend to be discarded. Evolutionary Algorithms [255] have been used in the deployment of Edge nodes and the offloading strategies. A subclass of Evolutionary Algorithms is the Genetic Algorithms (GA), which is characterized by the crossover principle in the reproduction of candidate solutions. GAs have been used for sequential task offloading and proactive fault tolerance [256]. Hybrid models, which combine GA with PSO, have also been proposed [257] and they achieve close to optimal task offloading of IoT applications, while minimizing the total makespan and energy consumption.

4.3.3. Constraint satisfaction methods

The task offloading problem has also been re-defined as a Constraint Satisfaction Problem (CSP) with multiple source of constraints such as SLA, QoS, QoE, the heterogeneity of devices, the particularities of VMs and the dynamicity of the task generation process. CSP [265] is related to the artificial intelligence operations research and aims to find feasible solutions by using methodologies such as constraint propagation, local search, backtracking and various heuristics. Specifically, task properties, user mobility and network constraints have been jointly formulated as a CSP [258], in order to reduce the task execution delay in Mobile Edge Computing infrastructures. A CSP formulation has also been used in combination with Min-conflicts scheduling algorithm [259], for achieving the necessary load balancing of the Edge resources and minimizing energy consumption. A more demanding offloading use case is the distributed processing of data streams. In this case, special emphasis is given to minimizing end-to-end latency through the appropriate placement of the stream operators, either on Cloud nodes or Edge devices. A CSP optimization framework has also been proposed in order to minimize this latency and satisfy the constraints of power, bandwidth and CPU utilization [260].

One prominent approach to address a CSP comes from the Constraint Programming (CP). CP [266] is a programming paradigm used in solving complex problems, where instead of defining a sequence of steps for the program to execute in order to obtain the result, one defines the relationships between variables in the form of constraints that must be met. Afterwards, by following the steps of branching and exploration, CP finds feasible solutions to the problem. CP has been proposed for a generic and easy-to-upgrade placement service for Fog Computing with short resolution times and quality solutions. Specifically, using Choco [261], a many times awarded constraint solver, we can estimate close to optimal solutions in terms of network infrastructure, applications graphs and metrics like the usage of storage, network and energy resources. In addition, CP has been combined with an event-based finite state model [262], in order to optimize mobile battery life and guarantee QoS and cost minimization simultaneously.

4.4. Control theory-based algorithms

Originally, Control Theory was designed to regulate the behavior of dynamic systems and keep the system output(s) following the desired

control signal, also called a reference. Control theory relies on feedback mechanisms and is widely applied to computing systems. Because of its nature to rely on feedback, which measures the difference between the actual output and the desired reference, control theory has been applied in the field of Edge Computing for implementing mechanisms of efficient decision-making control [267–269], network design selection [270,271], time-critical systems [75,76,272–275], admission control [276], network management [277], switching Edge [278] and network switching [279], among others. A summary of the control theory approaches, along with the objective of each study, the algorithm developed, type of offloading and the consideration of mobility, is listed in Table 4. Moreover, Fig. 5 illustrates the distribution of the key components of the existing control theory-based approaches.

4.4.1. Optimal control

The control theory foundations, specifically those based on linear optimal control theory (i.e., Linear Quadratic Regulator (LQR) [281]), consider the design of a selection strategy [270] in a heterogeneous wireless network, with the objective to maximize the network resource utilization, while meeting the constraints of the supported services. Linear controllers are designed to meet the system's constraints and QoS metrics [276]. The high efficiency of the LQR controllers guarantees the control performance of the system. Skarin et al. [280] consider the LQR for MIMO linear systems. Control theory can be deemed of utmost importance in UAVs-related task offloading where LQR-based controllers are used in order to achieve robust adaptive attitude control [282].

4.4.2. State feedback control

Another advantage of the control-theoretic approach in task offloading is that it provides a methodology for the modeling, analysis and evaluation of the system. Avgeris et al. [276] proposed a control theory approach to study the adaptive resource allocation problem for task offloading. The proposed two-level resource allocation and admission management system for an Edge application cluster, gives mobile users an alternative option for performing their tasks. The proposed controllers allow mobile users to offload application-specific tasks within the coverage area. However, it should be noted that mobility of users within the proximity of the cluster is not taken into consideration in this work. Kalatzis et al. [274] modeled the performance of IoT-based applications with a switched system and computed various equilibrium points that correspond to different operating conditions. Based on these points, a simple scaling mechanism was built to satisfy the varying workload demand. Extending [274], SMOKE [100] is a scalable resource allocation mechanism for UAV-based forest fire detection. UAVs are able to offload images to Edge servers for further processing. In the case of wildfire, the workload of the UAVs in the field increases significantly and the dynamic resource allocation is essential to achieve the desired QoS. A group of linear systems is used to model the container-based image processing services and a state-feedback controller is designed to scale each container's computing resources.

4.4.3. Lyapunov Optimization approaches

Lyapunov optimization algorithms provide a unique property of finding the sufficient conditions for stability in dynamical systems. Due to the stability theory of dynamical systems, Lyapunov-based optimization algorithms can be used in order to study the task offloading problem. In particular, for minimizing the energy consumption of mobile devices, there is a number of dynamic variables that need to be fine-tuned and converged to an optimal value in order to minimize the total energy consumption. Thus, Lyapunov optimization can be used to find the necessary stability in the CPU-cycle frequency of the device, transmission power, spectrum utilization and latency [139,170,178,191], while satisfying the necessary task execution constraints [115]. Another variable that dynamically fluctuates during task offloading is the resource availability. In this case, an online task offloading algorithm, which leverages Lyapunov optimization methods and utilizes the current system information, can be used to predict the user's resource availability [121]. The benefits are two-fold. First, network operators with global network information are trusted to make the comprehensive offloading decision for all the users. Second, the capabilities of mobile devices are constantly improving and the multiplexing advantage (due to the flexibility of resource availability between devices) can be exploited to enable the execution of collaborative tasks for a wide range of services.

4.4.4. Rest of the control approaches

This paragraph includes the control-based studies that cannot be classified in the previous categories. Dlamini et al. [277] developed an online algorithm for Edge network management based on predictive control. This control mechanism aims at optimizing a two-objective cost that includes energy consumption and QoS satisfaction. Sonmez et al. [158] proposed a fuzzy workload orchestrator for the Edge Computing environment. Here, a set of fuzzy rules assigns the offloaded requests to a computational unit in a hierarchical Edge Computing architecture. Spatharakis et al. [275] proposed a switching offloading mechanism for robotic applications (i.e., path planning and localization) within an Edge Computing setup. The offloading decision is based on the uncertainty of the mobile robot's position, the resource availability at the Edge servers and the complexity of the path planning.

In addition, control theory can be applied to address the task allocation problem [268] with a novel integrated Top-Down Bottom-Up (TDBU) approach. In particular, the top-down module (i) observes the bottom-up task preference decisions of the Edge devices and decides the optimal task offloading strategy to ensure the overall system performance; (ii) leverages top-down incentive schemes to implicitly guide the Edge devices to pick the tasks that are most likely to finish in time. Similarly, Wu et al. [269], proposed codeSpec, a decision making control theory approach (on the Edge device) that periodically renews its offloading decisions, at code level, with nearby IoT devices, in real-time. CodeSpec, shifts the destined devices from inter-domain servers to IoT devices nearby and only offloads binary code of user-specified regions across different instruction set architectures (e.g., x86_64, ARMv7 and IA-32), using control theory.

5. Open challenges and future directions

The concept of task offloading has evolved from the simple idea of migrating the computation intensive tasks of end-user devices at a remote location. The concept of Edge Computing and the arrival of new applications, enabled by recent trends in wireless communications such as the IoT and 5G, have introduced tremendous innovation opportunities for task offloading. However, new technical and business challenges arise. This section discusses future research directions and open issues in the context of task offloading.

5.1. Heterogeneous networks

A Heterogeneous Network (HetNet) consists of a macro cell layout with some possible Low Power Nodes (LPNs) placed throughout their coverage zones. Task offloading in HetNet is suited for three cases, as shown in Fig. 6: (i) Single-cell scenario, (ii) Contiguous cluster-cell scenario and (iii) Non-contiguous cluster-cell scenario.

In the context of single-cell scenarios, new offloading decision variables can be the interference and less congested cells selection. In the context of clustered-cells, the densest cell expands over several neighbor cells. The devices in the edge of the cell can extend the communication capacity (as well as other types of network management, e.g., energy or mobility) through nearby devices, to neighboring cells. In this case, the goal is to analyze the capability of a resource allocation

Table 4
Taxonomy of control theory-based task offloading algorithms.

Reference	Optimization objective					Algorithms				Mobility		Granularity		
	Delay	Energy	Bandwidth/ spectrum	Load balancing	Deployment cost	Optimal control	State feedback control	Lyapunov	Rest control approaches	Static	Mobile	Partial	Full	Edge–Cloud
[94]	✓	✓			✓			✓		✓			✓	
[100]		✓					✓			✓				✓
[115]	✓	✓						✓			✓	✓		
[121]	✓	✓	✓					✓		✓			✓	
[139]	✓	✓						✓		✓				✓
[158]			✓	✓	✓				✓		✓	✓		✓
[170]	✓	✓						✓		✓		✓		
[178]	✓	✓	✓					✓		✓				✓
[191]	✓	✓	✓					✓		✓		✓		
[268]	✓	✓							✓	✓		✓		
[269]	✓		✓				✓		✓	✓		✓		✓
[274]		✓				✓					✓	✓		
[275]	✓								✓		✓		✓	
[276]				✓		✓	✓			✓		✓		✓
[277]		✓							✓	✓		✓		
[280]	✓					✓				✓				✓

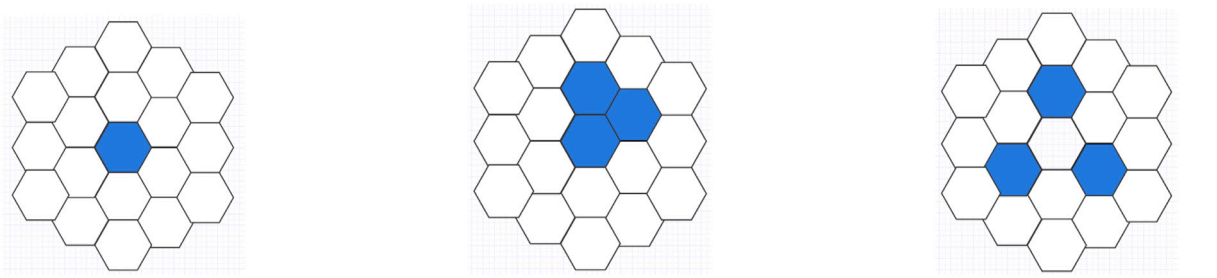


Fig. 6. Overview of general heterogeneous networks scenarios.

technique to shift to a remote Edge location, involving inter-cell communication and management aspects. A third scenario to be addressed concerns the communication of devices (small-cell environments) that are not adjacent, i.e., scenarios where the clustered cells may or may not be neighboring cells. This scenario assists in understanding how a resource allocation technique operates in terms of scalability, as well as in terms of supporting challenges such as long delays, or network partitions (as the small-cells are not in contiguous clusters).

Heterogeneous Dense Networks (HDHNs) consisting of clustered cells, require algorithms that are able to extend capacity over a distance of several cells towards the crowded cells. For the optimization of tasks with several small and distributed dense servers (either Edge or Cloud), however, the algorithms should draw capacity from cells within a short distance from the dense ones, such that only a few cells that are located close to the dense cells are affected. Initial studies in this area deal with simple scenarios of one end user and one Edge server in a single cell, or few end devices, one or more Edge servers and a central Cloud, while the results show the feasibility of task offloading with the combination of Edge and Cloud communications [41,200]. However, if multiple end-user devices reuse the spectrum to connect to multiple Edge and Cloud servers, imposing as a constraint to not degrade the QoE and the service continuity, the effect of signaling overhead, smooth handover and dynamic resource allocation on the offloading, becomes more significant. Such effects have not been thoroughly explored in the aforementioned studies.

5.2. Real-time distributed resource allocation

The optimization procedures during task offloading are primordial in order to handle crucial operations such as intelligent resource allocation and service continuity, by making independent and rational strategic decisions and smartly adapting to the environment. In this scenario, access networks are usually centralized, with all the traffic going through a central node (i.e., a BS). Furthermore, by offloading the tasks to a distributed Edge infrastructure, a separate backhaul connection is required which can increase the installation and energy costs for the mobile operators. Finding the right resources to offload the tasks to, in such distributed scenarios, is a critical objective, especially for heterogeneous networks. On top of that, the dynamic behavior of the user adds another level of complexity when the appropriate Edge site needs to be selected in order to achieve the task offloading objectives, such as low latency and resource optimization, while maintaining at the same time the user association information with the adjacent BS. Developing a real-time task offloading algorithm that considers the user interactions and a distributed Edge infrastructure, in order to improve the service delivery in dynamic scenarios, is one of the greatest currently open challenges. Moreover, in the heterogeneous networks, efficient real time allocation schemes that learn based on historic performance and adapt online to application's statistics, are still in their infancy.

5.3. Mobility-induced network dynamics

In many situations, the dynamic movement behavior of the users becomes the deciding factor on whether to offload the task or not. Even though few existing researches aim to take mobility into account, the particular case is still considered an open challenge. For example, developing algorithms by learning the user's behavior and network dynamics in parallel, in order to reduce communication and computational costs, are of utmost importance for new and emerging applications. Beyond 2020, there will be a growing demand for high user mobility applications such as drone-based applications, high speed trains, moving hotspots and 3D connectivity. Current solutions, would be difficult to be applied in such extremes scenarios, not only in terms

of accuracy but also in terms of minimum performance requirements (e.g over 500 km/h high speed mobility, high throughput and ultra low latency).

5.4. Node, resource and application heterogeneity

Another critical challenge is dealing the heterogeneity of the available infrastructure in terms of hardware and available resources. Both mobile and Edge devices are characterized by a great heterogeneity in terms of hardware, software and resource capabilities specifications. Furthermore, the existence of a large range of applications with different performance requirements, that are readily available at the same end device, can affect or limit the efficiency of the task offloading solution used. All these factors are key components during task offloading. Thus, maintaining adequate service delivery and service continuity, while addressing the task offloading in such a heterogeneous scenario, as a research topic, is still in its infancy.

5.5. Moving edge resources closer to the end devices

There exist several situations in which task offloading is indispensable but where the end devices are not able to directly offload data to an Edge server (e.g., that are not in range or do not have enough energy to reach it). In such cases, it may be useful to send off mobile Edge resources close to these end devices and adapt to their mobility and needs. However, obviously, mobile Edge resources might not be as powerful as stationary ones and might also be limited in terms of the services they can offer and their autonomy. Thus, the challenges imposed to task offloading here will be (i) how to trigger the sending of mobile resources, (ii) what kind of tasks to offload and (iii) from what end devices.

5.6. Security and privacy

Naturally, task offloading involves a huge amount of data outsourced to third party Edge infrastructures, thus security and privacy concepts are of paramount importance. These concepts can be addressed from different angles, i.e., (i) end user device, (ii) Edge data center and (iii) the actual data transmission over the network. Lately, a great increase in the variety of sophisticated attacks on end user devices has been observed, which constitutes the main target for attackers. Regarding the Edge infrastructures, threats are mainly focused on the data transmission between the different nodes of the network. Proposed solutions include various steganography and homomorphic encryption techniques, as well as hardware-based secure execution. However, when used individually, most of these solutions have limitations in their applications; e.g., encryption keys may be too large hence dramatically increasing the amount of transmitted and stored data, while computation on encrypted data is still in early research stages. Undoubtedly, Edge-related security and privacy threats are advancing in a quick manner, making it challenging to adapt to and deflect. Centralized monolithic security systems need to evolve as well into agile distributed solutions that combine more than one techniques, to fit better to the Edge Computing paradigm. Hence, task offloading solutions should be enhanced by taking into consideration security and confidentiality constraints.

5.7. Fault tolerance

Apart from security and privacy, an important factor contributing to building trust towards task offloading at the Edge is fault tolerance. As thoroughly discussed in the previous sections, mobility support is one of the most important requirements during task offloading and this is because autonomy of communication and freedom of movement are crucial criteria when it comes to users' satisfaction. Still, there are certain obstacles in achieving seamless connectivity and uninterrupted access to an Edge server while moving. For example, network bandwidth and data exchange rates may vary or connection might be lost. Thus, task offloading should be enhanced with fault tolerance techniques to guarantee the successful transmission and execution of the task, as well as minimize application response time and energy consumption in end-user devices.

5.8. Control-related challenges

Although control theory is widely applied in Cloud elasticity and resource allocation problems [283], there are still open challenges on task offloading and Edge Computing that can be addressed by control techniques. Apart from respecting the QoS requirements, control theory is able to guarantee important system properties, e.g., stability, positive invariant sets and ultimate boundedness [284], against the inherent system's uncertainties and external disturbances. Intermittent connectivity, the innate management features of the virtualization technologies and the limited resources at the edge of the network, lead to a highly volatile dynamic environment that necessitates advanced modeling and control methodologies. Regarding the modeling of the offloading-based applications, control theory provides many modeling alternatives involving switching systems [145,284], Linear Parameter Varying (LPV) systems [285,286] and Fuzzy Takagi–Sugeno systems [287,288], that allow the natural incorporation of uncertainties and disturbances in the performance model.

5.9. Controller design for cyber–physical systems (CPS)

Another active and very practical challenge in the context of IoT and mobile-enabled computing, is the controller design for cyber–physical systems (CPS), found e.g., in manufacturing, transportation and collaborative robotics. In the context of dynamic networks and remote computing, a joint co-design decision making strategy for task offloading, resource allocation and controller design for CPS, is more appropriate when compared to separate layers of controllers for the infrastructure resources and the system to be controlled. This new generation of controllers will be made possible by merging two sets of models, namely (a) the performance model described above and (b) the process model (having, for example, variables related to position, orientation and velocity of mobile devices, lighting conditions, room temperature and mode of operation of sensors). The co-designed controllers should address many of the non-idealities of the dynamic networks found during task offloading, where resources must be used parsimoniously, in balance with the constraints and the overall objective [289]. It is worth mentioning that in control engineering, shared and imperfect communication networks between the controller and the sensor/actuator have been studied extensively, generating the branch of Networked Control Systems (NCS) [290]. Several developed methods address time delays and packet dropouts of NCS, utilizing perturbation theory, Lyapunov stability theory and hybrid systems analysis, including probabilistic methods involving Markov chains and stochastic automata [291–293]. A breakthrough will be the emergence of event-triggered and self-triggered control, that allows asynchronous sampling, thus reducing the network traffic, while at the same time providing guaranteed trade-offs of the degradation of the closed-loop system performance [294]. Consequently, in the context of task offloading, a timely challenge is to provide co-design control formulations that develop dynamic task offloading as well as control design algorithms for CPS, taking simultaneously into account the schedulability, available and requested network resources, Edge resources and energy consumption. It is anticipated that such control algorithms will improve performance, utilization of the underlying infrastructure as well as resilience and robustness of the systems to be controlled.

6. Conclusion

In this paper, we have presented a detailed and comprehensive study of the task offloading problem and we have extensively analyzed the main components and different computing paradigms of an end-to-end communication path. This path starts at the end device (i.e., mobile/IoT devices) and leverages the benefits of the added computational resources at the edge of the network, before ending up to the Cloud. Nonetheless, offloading parts of an application from an end device to a remote Edge or Cloud site, arises a number of challenges mostly related to the dynamic network behavior and the resource allocation problem to be solved. Between these challenges, the type of end devices in terms of mobility is expected to be one of the main characteristics of task offloading that can dictate the scheduling and allocation of the offloaded tasks. Prominent solutions addressing these challenges have given the necessary momentum to task offloading in order to be considered as a viable solution not only to existing, but to emerging and upcoming applications such as immersive applications, autonomous vehicles and robotics.

The benefits of task offloading are numerous, allowing to increase the QoS and QoE of the application, while extending the battery lifetime of the end devices. To achieve that, specific emphasis is given on the optimization models used along with the different objectives, in order to partition and allocate the tasks of end devices into an end-to-end communication path that creates a user to Cloud continuum. Since this problem contains various dynamic parameters in terms of network and user behavior, appropriate AI and ML techniques to anticipate traffic demands have also been presented. Control theory techniques have also been investigated as an alternative way to address the uncertainties of this dynamic problem, with the aim to ensure the necessary stability of the task offloading systems. Our survey concluded with some interesting open challenges that will shape and transform the task offloading problem for future network scenarios and applications. These future directions emphasize on control co-design, dynamic and real-time allocation in a heterogeneous Edge environment, and secure and highly mobile network platforms.

CRediT authorship contribution statement

Firdose Saeik: Conceptualization, Investigation, Writing - original draft, Writing - review & editing. **Marios Avgeris:** Conceptualization, Investigation, Writing - original draft, Writing - review & editing. **Dimitrios Spatharakis:** Conceptualization, Investigation, Writing - original draft, Writing - review & editing. **Nina Santi:** Conceptualization, Investigation, Writing - original draft, Writing - review & editing. **Dimitrios Dechouniotis:** Conceptualization, Investigation, Writing - original draft, Writing - review & editing. **John Violos:** Conceptualization, Investigation, Writing - original draft, Writing - review & editing. **Aris Leivadeas:** Conceptualization, Investigation, Writing - original draft, Writing - review & editing. **Nikolaos Athanasopoulos:** Conceptualization, Investigation, Writing - original draft, Writing - review & editing. **Nathalie Mitton:** Conceptualization, Investigation, Writing - original draft, Writing - review & editing. **Symeon Papavassiliou:** Conceptualization, Investigation, Writing - original draft, Writing - review & editing.

Declaration of competing interest

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

Acknowledgment

This work was supported in part by the CHIST-ERA-2018-DRUID-NET project, France.

References

[1] A. Gupta, R.K. Jha, A survey of 5G network: Architecture and emerging technologies, IEEE Access 3 (2015) 1206–1232.
[2] M. Series, IMT Vision–framework and overall objectives of the future development of IMT for 2020 and beyond, Recomm. ITU (2015) 2083–0.
[3] M. Altamimi, A. Abdrabou, K. Naik, A. Nayak, Energy cost models of smartphones for task offloading to the cloud, IEEE Trans. Emerg. Top. Comput. 3 (3) (2015) 384–398.
[4] X. Ma, Y. Zhao, L. Zhang, H. Wang, L. Peng, When mobile terminals meet the cloud: computation offloading as the bridge, IEEE Netw. 27 (5) (2013) 28–33.
[5] W. Yu, F. Liang, X. He, W.G. Hatcher, C. Lu, J. Lin, X. Yang, A survey on the edge computing for the Internet of Things, IEEE Access 6 (2017) 6900–6919.
[6] P. Mach, Z. Becvar, Mobile edge computing: A survey on architecture and computation offloading, IEEE Commun. Surv. Tutor. 19 (3) (2017) 1628–1656.
[7] C. Jiang, X. Cheng, H. Gao, X. Zhou, J. Wan, Toward computation offloading in edge computing: A survey, IEEE Access 7 (2019) 131543–131558.
[8] Y. Mao, C. You, J. Zhang, K. Huang, K.B. Letaief, A survey on mobile edge computing: The communication perspective, IEEE Commun. Surv. Tutor. 19 (4) (2017) 2322–2358.
[9] C.-H. Hong, B. Varghese, Resource management in fog/edge computing: a survey on architectures, infrastructure, and algorithms, ACM Comput. Surv. 52 (5) (2019) 1–37.
[10] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, W. Wang, A survey on mobile edge networks: Convergence of computing, caching and communications, IEEE Access 5 (2017) 6757–6779.
[11] T.K. Rodrigues, K. Suto, H. Nishiyama, J. Liu, N. Kato, Machine learning meets computation and communication control in evolving edge and cloud: Challenges and future perspective, IEEE Commun. Surv. Tutor. (2019).
[12] J. Wang, J. Pan, F. Esposito, P. Calyam, Z. Yang, P. Mohapatra, Edge cloud offloading algorithms: Issues, methods, and perspectives, ACM Comput. Surv. 52 (1) (2019) 1–23.
[13] Q. Pham, F. Fang, V.N. Ha, M.J. Piran, M. Le, L.B. Le, W. Hwang, Z. Ding, A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art, IEEE Access 8 (2020) 116974–117017.
[14] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, J.P. Jue, All one needs to know about fog computing and related edge computing paradigms: A complete survey, J. Syst. Archit. 98 (2019) 289–330.
[15] L. Lin, X. Liao, H. Jin, P. Li, Computation offloading toward edge computing, Proc. IEEE 107 (8) (2019) 1584–1607.
[16] X. Shan, H. Zhi, P. Li, Z. Han, A survey on computation offloading for mobile edge computing information, in: 2018 IEEE 4th International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing, (HPSC) and IEEE International Conference on Intelligent Data and Security (IDS), IEEE, 2018, pp. 248–251.
[17] Y. Ai, M. Peng, K. Zhang, Edge computing technologies for Internet of Things: a primer, Digit. Commun. Netw. 4 (2) (2018) 77–86.
[18] W.Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, A. Ahmed, Edge computing: A survey, Future Gener. Comput. Syst. 97 (2019) 219–235.
[19] L. Mendiboure, M.-A. Chalouf, F. Krief, Edge computing based applications in vehicular environments: Comparative study and main issues, J. Comput. Sci. Tech. 34 (4) (2019) 869–886.
[20] P. Mell, T. Grance, et al., The NIST Definition of Cloud Computing, Computer Security Division, National Information Technology Laboratory, 2011.
[21] L. Mei, W.K. Chan, T. Tse, A tale of clouds: Paradigm comparisons and some thoughts on research issues, in: 2008 IEEE Asia-Pacific Services Computing Conference, IEEE, 2008, pp. 464–469.
[22] N. Fernando, S.W. Loke, W. Rahayu, Dynamic mobile cloud computing: Ad hoc and opportunistic job sharing, in: 2011 Fourth IEEE International Conference on Utility and Cloud Computing, IEEE, 2011, pp. 281–286.
[23] G. Huerta-Canepa, D. Lee, A virtual cloud computing provider for mobile devices, in: Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and beyond, ACM, 2010, p. 6.
[24] A.U.R. Khan, M. Othman, S.A. Madani, S.U. Khan, A survey of mobile cloud computing application models, IEEE Commun. Surv. Tutor. 16 (1) (2014) 393–413.
[25] N.I.M. Enzai, M. Tang, A taxonomy of computation offloading in mobile cloud computing, in: 2014 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, 2014, pp. 19–28.
[26] M. Satyanarayanan, V. Bahl, R. Caceres, N. Davies, The case for vm-based cloudlets in mobile computing, IEEE Pervas. Comput. (2009).
[27] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the Internet of Things, in; Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, 2012, pp. 13–16.
[28] S. Antonio, Cisco delivers vision of fog computing to accelerate value from billions of connected devices, Cisco (2014).
[29] O. Consortium, et al., Openfog reference architecture for fog computing, Archit. Work. Group (2017).

[30] R. Cisco, M.Y. Upc, M. Nemirovsky, Fog computing, in: Proc. Cloud Assist. Serveys Eur. Conf. Bled, 2012, pp. 1–15.

[31] F. Bonomi, R. Milito, P. Natarajan, J. Zhu, Fog computing: A platform for Internet of Things and analytics, in: Big Data and Internet of Things: A Roadmap for Smart Environments, Springer, 2014, pp. 169–186.

[32] M. Chiang, T. Zhang, Fog and IoT: An overview of research opportunities, IEEE Internet Things J. 3 (6) (2016) 854–864.

[33] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, S. Chen, Vehicular fog computing: A viewpoint of vehicles as the infrastructures, IEEE Trans. Veh. Technol. 65 (6) (2016) 3860–3873.

[34] S.A. Soleymani, A.H. Abdullah, M. Zareei, M.H. Anisi, C. Vargas-Rosales, M.K. Khan, S. Goudarzi, A secure trust model based on fuzzy logic in vehicular ad hoc networks with fog computing, IEEE Access 5 (2017) 15619–15629.

[35] M.S. Elbamby, M. Bennis, W. Saad, Proactive edge computing in latency-constrained fog networks, in: 2017 European Conference on Networks and Communications, EuCNC, IEEE, 2017, pp. 1–6.

[36] D. You, T.V. Doan, R. Torre, M. Mehrabi, A. Kropp, V. Nguyen, H. Salah, G.T. Nguyen, F.H. Fitzek, Fog computing as an enabler for immersive media: Service scenarios and research opportunities, IEEE Access 7 (2019) 65797–65810.

[37] M. Iorga, L. Feldman, R. Barton, M. Martin, N. Goren, C. Mahmoudi, The Nist Definition of Fog Computing, Tech. Rep., National Institute of Standards and Technology, 2017.

[38] Y.C. Hu, M. Patel, D. Sabella, N. Sprecher, V. Young, Mobile edge computing—A key technology towards 5G, ETSI White Paper 11 (11) (2015) 1–16.

[39] M. ETSI, Mobile edge computing (mec); framework and reference architecture, in: ETSI, DGS MEC, vol. 3, 2016.

[40] D. Sabella, A. Vaillant, P. Kuure, U. Rauschenbach, F. Giust, Mobile-edge computing architecture: The role of MEC in the Internet of Things, IEEE Consumer Electron. Mag. 5 (4) (2016) 84–91.

[41] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, L. Pan, S. Maharjan, Y. Zhang, Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks, IEEE Access 4 (2016) 5896–5907.

[42] I. Ketykó, L. Kecskés, C. Nemes, L. Farkas, Multi-user computation offloading as multiple knapsack problem for 5G mobile edge computing, in: 2016 European Conference on Networks and Communications, EuCNC, IEEE, 2016, pp. 225–229.

[43] K. Zhang, Y. Mao, S. Leng, A. Vinel, Y. Zhang, Delay constrained offloading for mobile edge computing in cloud-enabled vehicular networks, in: 2016 8th International Workshop on Resilient Networks Design and Modeling, RNDM, IEEE, 2016, pp. 288–294.

[44] T. Taleb, S. Dutta, A. Ksentini, M. Iqbal, H. Flinck, Mobile edge computing potential in making cities smarter, IEEE Commun. Mag. 55 (3) (2017).

[45] T.X. Tran, M.-P. Hosseini, D. Pompili, Mobile edge computing: Recent efforts and five key research directions, IEEE COMSOC MMTC Commun.-Frontiers (2017).

[46] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, D. Sabella, On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration, IEEE Commun. Surv. Tutor. 19 (3) (2017) 1657–1681.

[47] S. Kekki, W. Featherstone, Y. Fang, P. Kuure, A. Li, A. Ranjan, D. Purkayastha, F. Jiangping, D. Frydman, G. Verin, et al., MEC In 5G networks, ETSI White Paper 28 (2018) 1–28.

[48] V. Bahl, Emergence of micro datacenter (cloudlets/edges) for mobile computing, in: Microsoft Devices & Networking Summit 2015, 2015.

[49] European Telecommunication Standards Institute (ETSI), Multi-access Edge Computing (MEC), https://www.etsi.org/technologies/multi-access-edge-computing.

[50] E. Miluzzo, T. Wang, A.T. Campbell, Eyephone: activating mobile phones with your eyes, in: Proceedings of the Second ACM SIGCOMM Workshop on Networking, Systems, and Applications on Mobile Handhelds, ACM, 2010, pp. 15–20.

[51] R. Kemp, N. Palmer, T. Kielmann, H. Bal, Cuckoo: a computation offloading framework for smartphones, in: International Conference on Mobile Computing, Applications, and Services, Springer, 2010, pp. 59–79.

[52] R. Kemp, N. Palmer, T. Kielmann, H. Bal, Opportunistic communication for multiplayer mobile gaming: Lessons learned from photoshoot, in: Proceedings of the Second International Workshop on Mobile Opportunistic Networking, ACM, 2010, pp. 182–184.

[53] S. Bohez, J. De Turck, T. Verbelen, P. Simoens, B. Dhoedt, Mobile, collaborative augmented reality using cloudlets, in: 2013 International Conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications, IEEE, 2013, pp. 45–54.

[54] T. Verbelen, P. Simoens, F. De Turck, B. Dhoedt, Leveraging cloudlets for immersive collaborative applications, IEEE Pervas. Comput. 12 (4) (2013) 30–38.

[55] J.-M. Chung, Y.-S. Park, J.-H. Park, H. Cho, Adaptive cloud offloading of augmented reality applications on smart devices for minimum energy consumption, Ksii Trans. Internet Inf. Syst. 9 (8) (2015).

[56] U. Drolia, R. Martins, J. Tan, A. Chheda, M. Sanghavi, R. Gandhi, P. Narasimhan, The case for mobile edge-clouds, in: 2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing, IEEE, 2013, pp. 209–215.

[57] J. Dolezal, Z. Becvar, T. Zeman, Performance evaluation of computation offloading from mobile device to the edge of mobile network, in: 2016 IEEE Conference on Standards for Communications and Networking, CSCN, IEEE, 2016, pp. 1–7.

[58] I.-S. Comşa, G.-M. Muntean, R. Trestian, An innovative machine-learning-based scheduling solution for improving live UHD video streaming quality in highly dynamic network environments, IEEE Trans. Broadcast. (2020).

[59] S. Mangiante, G. Klas, A. Navon, Z. GuanHua, J. Ran, M.D. Silva, VR is on the edge: How to deliver 360 videos in mobile networks, in: Proceedings of the Workshop on Virtual Reality and Augmented Reality Network, 2017, pp. 30–35.

[60] J. Chakareski, VR/AR immersive communication: Caching, edge computing, and transmission trade-offs, in: Proceedings of the Workshop on Virtual Reality and Augmented Reality Network, 2017, pp. 36–41.

[61] Q. Liu, S. Huang, J. Opadere, T. Han, An edge network orchestrator for mobile augmented reality, in: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, IEEE, 2018, pp. 756–764.

[62] E. Fraedrich, R. Cyganski, I. Wolf, B. Lenz, User Perspectives on Autonomous Driving: A Use-Case-Driven Study in Germany, Geographisches Institut, Humboldt-Universität zu Berlin, 2016.

[63] U. Puetzschler, LTE and Car2x: Connected cars on the way to 5G, in: Mobile Broadband SIG, vol. 6, 2016.

[64] Y. Sahni, J. Cao, S. Zhang, L. Yang, Edge mesh: A new paradigm to enable distributed intelligence in Internet of Things, IEEE Access 5 (2017) 16441–16458.

[65] X. Wang, Z. Ning, L. Wang, Offloading in internet of vehicles: A fog-enabled real-time traffic management system, IEEE Trans. Ind. Inf. 14 (10) (2018) 4568–4578.

[66] G. Hu, W.P. Tay, Y. Wen, Cloud robotics: architecture, challenges and applications, IEEE Netw. 26 (3) (2012) 21–28.

[67] B. Kehoe, S. Patil, P. Abbeel, K. Goldberg, A survey of research on cloud robotics and automation, IEEE Trans. Autom. Sci. Eng. 12 (2) (2015) 398–409.

[68] S. Jordan, T. Haidegger, L. Kovács, I. Felde, I. Rudas, The rising prospects of cloud robotic applications, in: 2013 IEEE 9th International Conference on Computational Cybernetics, ICCC, IEEE, 2013, pp. 327–332.

[69] D. Song, A.K. Tanwani, K. Goldberg, B. Siciliano, Networked-, Cloud-and Fog-Robotics, Springer, 2019.

[70] A.K. Tanwani, R. Anand, J.E. Gonzalez, K. Goldberg, RILaaS: Robot Inference and Learning as a Service, IEEE Robotics Autom. Lett. (2020).

[71] K. Bekris, R. Shome, A. Krontiris, A. Dobson, Cloud automation: Precomputing roadmaps for flexible manipulation, IEEE Robot. Autom. Mag. 22 (2) (2015) 41–50.

[72] J. Glover, D. Rus, N. Roy, Probabilistic models of object geometry for grasp planning, in: Proceedings of Robotics: Science and Systems IV, Zurich, Switzerland, 2008, pp. 278–285.

[73] L. Riazuelo, J. Civera, J.M. Montiel, C2tam: A cloud framework for cooperative tracking and mapping, Robot. Auton. Syst. 62 (4) (2014) 401–413.

[74] L. Turnbull, B. Samanta, Cloud robotics: Formation control of a multi robot system utilizing cloud infrastructure, in: 2013 Proceedings of IEEE Southeastcon, IEEE, 2013, pp. 1–4.

[75] G. Mohanarajah, V. Usenko, M. Singh, R. D'Andrea, M. Waibel, Cloud-based collaborative 3D mapping in real-time with low-cost robots, IEEE Trans. Autom. Sci. Eng. 12 (2) (2015) 423–431.

[76] N. Tian, A.K. Tawani, K. Goldberg, S. Sojoudi, Motion segmentation and synthesis for latency mitigation in a cloud robotic tele-operation system, in: International Symposium on Robotics Research, 2019.

[77] S.L. Bowman, N. Atanasov, K. Daniilidis, G.J. Pappas, Probabilistic data association for semantic slam, in: 2017 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2017, pp. 1722–1729.

[78] L. Riazuelo, M. Tenorth, D. Marco, M. Salas, L. Mosenlechner, L. Kunze, M. Beetz, J. Tardos, L. Montano, J. Montiel, Roboearth web-enabled and knowledge-based active perception, in: IROS Workshop on AI-Based Robotics, 2013.

[79] A.K. Tanwani, N. Mor, J. Kubiatowicz, J.E. Gonzalez, K. Goldberg, A fog robotics approach to deep robot learning: Application to object recognition and grasp planning in surface decluttering, in: 2019 International Conference on Robotics and Automation, ICRA, IEEE, 2019, pp. 4559–4566.

[80] A.W. Services, AWS robomaker [online], 2021, https://aws.amazon.com/robomaker/. (Accessed 20 January 2021).

[81] Google, Cloud robotics core [online], 2021, https://googlecloudrobotics.github.io/core/, (Accessed 20 January 2021).

[82] Rapyuta robotics [online], 2021, https://www.rapyuta-robotics.com. (Accessed 20 January 2021).

[83] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, B. Weihl, Globally distributed content delivery, IEEE Internet Comput. 6 (5) (2002) 50–58.

[84] C. Papagianni, A. Leivadeas, S. Papavassiliou, A cloud-oriented content delivery network paradigm: Modeling and assessment, IEEE Trans. Dependable Secure Comput. 10 (5) (2013) 287–300.

[85] K. Bilal, A. Erbad, M. Hefeeda, Crowdsourced multi-view live video streaming using cloud computing, IEEE Access 5 (2017) 12635–12647.

[86] J.O. Fajardo, I. Taboada, F. Liberal, Improving content delivery efficiency through multi-layer mobile edge adaptation, IEEE Netw. 29 (6) (2015) 40–46.

[87] T.X. Tran, P. Pandey, A. Hajisami, D. Pompili, Collaborative multi-bitrate video caching and processing in mobile-edge computing networks, in: 2017 13th Annual Conference on Wireless on-Demand Network Systems and Services, WONS, IEEE, 2017, pp. 165–172.

[88] J. Ren, G. Yu, Y. Cai, Y. He, Latency optimization for resource allocation in mobile-edge computation offloading, IEEE Trans. Wireless Commun. 17 (8) (2018) 5506–5519.

[89] Y.-H. Kao, B. Krishnamachari, M.-R. Ra, F. Bai, Hermes: Latency optimal task assignment for resource-constrained mobile computing, IEEE Trans. Mob. Comput. 16 (11) (2017) 3056–3069.

[90] S. Yang, Y. He, X. Zheng, Fovr: Attention-based VR streaming through bandwidth-limited wireless networks, in: 2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking, SECON, IEEE, 2019, pp. 1–9.

[91] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of Things (IoT): A vision, architectural elements, and future directions, Future Gener. Comput. Syst. 29 (7) (2013) 1645–1660.

[92] I. Farris, L. Militano, M. Nitti, L. Atzori, A. Iera, Federated edge-assisted mobile clouds for service provisioning in heterogeneous IoT environments, in: 2015 IEEE 2nd World Forum on Internet of Things, WF-IoT, IEEE, 2015, pp. 591–596.

[93] C. Zhu, X. Li, L. Song, L. Xiang, Development of a theoretically based thermal model for lithium ion battery pack, J. Power Sources 223 (2013) 155–164.

[94] Y. Nan, W. Li, W. Bao, F.C. Delicato, P.F. Pires, Y. Dou, A.Y. Zomaya, Adaptive energy-aware computation offloading for cloud of things systems, IEEE Access 5 (2017) 23947–23957.

[95] I. Farris, L. Militano, M. Nitti, L. Atzori, A. Iera, Mifaas: A mobile-IoT-federation-as-a-service model for dynamic cooperation of IoT cloud providers, Future Gener. Comput. Syst. 70 (2017) 126–137.

[96] S.K. Sharma, X. Wang, Live data analytics with collaborative edge and cloud processing in wireless IoT networks, IEEE Access 5 (2017) 4621–4635.

[97] A. Yousefpour, G. Ishigaki, R. Gour, J.P. Jue, On reducing IoT service delay via fog offloading, IEEE Internet Things J. 5 (2) (2018) 998–1010.

[98] K. Kim, C.S. Hong, Optimal task-UAV-edge matching for computation offloading in uav assisted mobile edge computing, in: 2019 20th Asia-Pacific Network Operations and Management Symposium, APNOMS, IEEE, 2019, pp. 1–4.

[99] S. Chen, Q. Wang, J. Chen, T. Wu, An intelligent task offloading algorithm (iTOA) for UAV network, in: 2019 IEEE Globecom Workshops, GC Wkshps, IEEE, 2019, pp. 1–6.

[100] M. Avgeris, D. Spatharakis, D. Dechouniotis, N. Kalatzis, I. Roussaki, S. Papavassiliou, Where there is fire there is smoke: a scalable edge computing framework for early fire detection, Sensors 19 (3) (2019) 639.

[101] J. Wang, M. Conrad Meyer, Y. Wu, Y. Wang, Maximum data-resolution efficiency for fog-computing supported spatial big data processing in disaster scenarios, IEEE Trans. Parallel Distrib. Syst. 30 (2019) 1826–1842.

[102] J. Zhou, J. Fan, J. wang, J. Zhu, Task offloading for social sensing applications in mobile edge computing, in: 2019 Seventh International Conference on Advanced Cloud and Big Data, CBD, IEEE, 2019, pp. 333–338.

[103] G. Ananthanarayanan, P. Bahl, P. Bodík, K. Chintalapudi, M. Philipose, L. Ravindranath, S. Sinha, Real-time video analytics: The killer app for edge computing, Computer 50 (10) (2017) 58–67.

[104] J. Wang, J. Pan, F. Esposito, P. Calyam, Z. Yang, P. Mohapatra, Edge cloud offloading algorithms: Issues, methods, and perspectives, ACM Comput. Surv. 52 (1) (2019) 2:1–2:23, http://doi.acm.org/10.1145/3284387.

[105] T.X. Tran, A. Hajisami, P. Pandey, D. Pompili, Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges, IEEE Commun. Mag. 55 (4) (2017) 54–61.

[106] J. Violos, S. Pelekis, A. Berdelis, S. Tsanakas, K. Tserpes, T. Varvarigou, Predicting visitor distribution for large events in smart cities, in: 2019 IEEE International Conference on Big Data and Smart Computing, BigComp, 2019, pp. 1–8 (ISSN: 2375-9356).

[107] T.X. Tran, D. Pompili, Joint task offloading and resource allocation for multi-server mobile-edge computing networks, IEEE Trans. Veh. Technol. 68 (1) (2018) 856–868.

[108] J. Violos, E. Psomakelis, K. Tserpes, F. Aisopos, T. Varvarigou, Leveraging user mobility and mobile app services behavior for optimal edge resource utilization, in: Proceedings of the International Conference on Omni-Layer Intelligent Systems, COINS '19, Association for Computing Machinery, Crete, Greece, 2019, pp. 7–12.

[109] L. Yang, H. Zhang, M. Li, J. Guo, H. Ji, Mobile edge computing empowered energy efficient task offloading in 5G, IEEE Trans. Veh. Technol. 67 (7) (2018) 6398–6409.

[110] Z. Xu, X. Liu, G. Jiang, B. Tang, A time-efficient data offloading method with privacy preservation for intelligent sensors in edge computing, EURASIP J. Wireless Commun. Networking 2019 (1) (2019) 1–12.

[111] F. Liu, Z. Huang, L. Wang, Energy-efficient collaborative task computation offloading in cloud-assisted edge computing for IoT sensors, Sensors 19 (5) (2019) 1105.

[112] M. Chen, B. Liang, M. Dong, Multi-user multi-task offloading and resource allocation in mobile cloud systems, IEEE Trans. Wireless Commun. 17 (10) (2018) 6790–6805.

[113] Y. Cao, T. Jiang, C. Wang, Optimal radio resource allocation for mobile task offloading in cellular networks, IEEE Netw. 28 (5) (2014) 68–73.

[114] Y. Cao, C. Long, T. Jiang, S. Mao, Share communication and computation resources on mobile devices: a social awareness perspective, IEEE Wirel. Commun. 23 (4) (2016) 52–59.

[115] D. Huang, P. Wang, D. Niyato, A dynamic offloading algorithm for mobile computing, IEEE Trans. Wireless Commun. 11 (6) (2012) 1991–1995.

[116] S. Sardellitti, S. Barbarossa, G. Scutari, Distributed mobile cloud computing: Joint optimization of radio and computational resources, in: 2014 IEEE Globecom Workshops, GC Wkshps, IEEE, 2014, pp. 1505–1510.

[117] O. Munoz, A. Pascual-Iserte, J. Vidal, Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading, IEEE Trans. Veh. Technol. 64 (10) (2014) 4738–4755.

[118] J. Plachy, Z. Becvar, E.C. Strinati, Dynamic resource allocation exploiting mobility prediction in mobile edge computing, in: 2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications, PIMRC, IEEE, 2016, pp. 1–6.

[119] T. Truong-Huu, C.-K. Tham, D. Niyato, To offload or to wait: An opportunistic offloading algorithm for parallel tasks in a mobile cloud, in: 2014 IEEE 6th International Conference on Cloud Computing Technology and Science, IEEE, 2014, pp. 182–189.

[120] M. Chen, Y. Hao, Y. Li, C.-F. Lai, D. Wu, On the computation offloading at ad hoc cloudlet: architecture and service modes, IEEE Commun. Mag. 53 (6) (2015) 18–24.

[121] L. Pu, X. Chen, J. Xu, X. Fu, D2D fogging: An energy-efficient and incentive-aware task offloading framework via network-assisted D2D collaboration, IEEE J. Sel. Areas Commun. 34 (12) (2016) 3887–3901.

[122] S. Zhou, Y. Sun, Z. Jiang, Z. Niu, Exploiting moving intelligence: Delay-optimized computation offloading in vehicular fog networks, IEEE Commun. Mag. 57 (5) (2019) 49–55.

[123] W. Li, X. You, Y. Jiang, J. Yang, L. Hu, Opportunistic computing offloading in edge clouds, J. Parallel Distrib. Comput. 123 (2019) 69–76.

[124] Y. Sun, S. Zhou, J. Xu, EMM: Energy-aware mobility management for mobile edge computing in ultra dense networks, IEEE J. Sel. Areas Commun. 35 (11) (2017) 2637–2646.

[125] Z. Wang, Z. Zhao, G. Min, X. Huang, Q. Ni, R. Wang, User mobility aware task assignment for mobile edge computing, Future Gener. Comput. Syst. 85 (2018) 1–8.

[126] G. Ahani, D. Yuan, BS-assisted task offloading for D2D networks with presence of user mobility, in: 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring), IEEE, 2019, pp. 1–5.

[127] Z. Ning, P. Dong, X. Wang, J.J. Rodrigues, F. Xia, Deep reinforcement learning for vehicular edge computing: An intelligent offloading system, ACM Trans. Intell. Syst. Technol. (TIST) 10 (6) (2019) 1–24.

[128] T.M.T. Do, O. Dousse, M. Miettinen, D. Gatica-Perez, A probabilistic kernel method for human mobility prediction with smartphones, Pervasive Mob. Comput. 20 (2015) 13–28.

[129] B. Li, H. Zhang, H. Lu, User mobility prediction based on Lagrange's interpolation in ultra-dense networks, in: 2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications, PIMRC, IEEE, 2016, pp. 1–6.

[130] H.A. Alameddine, S. Sharafeddine, S. Sebbah, S. Ayoubi, C. Assi, Dynamic task offloading and scheduling for low-latency IoT services in multi-access edge computing, IEEE J. Sel. Areas Commun. 37 (3) (2019) 668–682.

[131] S. Wu, W. Xia, W. Cui, Q. Chao, Z. Lan, F. Yan, L. Shen, An efficient offloading algorithm based on support vector machine for mobile edge computing in vehicular networks, in: 2018 10th International Conference on Wireless Communications and Signal Processing, WCSP, IEEE, 2018, pp. 1–6.

[132] M. Maanoja, M. Weckström, Location services, Google Patents, US Patent 7, 069, 023, 2006.

[133] S. Deng, L. Huang, D. Hu, J.L. Zhao, Z. Wu, Mobility-enabled service selection for composite services, IEEE Trans. Serv. Comput. 9 (3) (2016) 394–407.

[134] S. Deng, L. Huang, J. Taheri, J. Yin, M. Zhou, A.Y. Zomaya, Mobility-aware service composition in mobile communities, IEEE Trans. Syst. Man Cybern.: Syst. 47 (3) (2017) 555–568.

[135] S. Gambs, M.-O. Killijian, M.N. del Prado Cortez, Next place prediction using mobility Markov chains, in: Proceedings of the First Workshop on Measurement, Privacy, and Mobility, 2012, pp. 1–6.

[136] K. Lee, I. Shin, User mobility model based computation offloading decision for mobile cloud, J. Comput. Sci. Eng. 9 (3) (2015) 155–162.

[137] X. Sun, N. Ansari, Adaptive avatar handoff in the cloudlet network, IEEE Trans. Cloud Comput. 7 (3) (2019) 664–676.

[138] M. Chen, Y. Hao, M. Qiu, J. Song, D. Wu, I. Humar, Mobility-aware caching and computation offloading in 5G ultra-dense cellular networks, Sensors 16 (7) (2016) 974.

[139] J. Xu, L. Chen, P. Zhou, Joint service caching and task offloading for mobile edge computing in dense networks, in: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, IEEE, 2018, pp. 207–215.

[140] D. Wang, Z. Liu, X. Wang, Y. Lan, Mobility-aware task offloading and migration schemes in fog computing networks, IEEE Access 7 (2019) 43356–43368.

[141] Y. Shi, S. Chen, X. Xu, MAGA: A mobility-aware computation offloading decision for distributed mobile cloud computing, IEEE Internet Things J. 5 (1) (2017) 164–174.

[142] F. Yu, H. Chen, J. Xu, DMPO: Dynamic mobility-aware partial offloading in mobile edge computing, Future Gener. Comput. Syst. 89 (2018) 722–735.

[143] V.A. Siris, D. Kalyvas, Enhancing mobile data offloading with mobility prediction and prefetching, ACM SIGMOBILE Mob. Comput. Commun. Rev. 17 (1) (2013) 22–29.

[144] C. Yang, Y. Liu, X. Chen, W. Zhong, S. Xie, Efficient mobility-aware task offloading for vehicular edge computing networks, IEEE Access 7 (2019) 26652–26664.

[145] D. Spatharakis, I. Dimolitsas, D. Dechouniotis, G. Papathanail, I. Fotoglou, P. Papadimitriou, S. Papavassiliou, A scalable edge computing architecture enabling smart offloading for location based services, Pervasive Mob. Comput. 67 (2020) 101217.

[146] G. Papathanail, I. Fotoglou, C. Demertzis, A. Pentelas, K. Sgouromitis, P. Papadimitriou, D. Spatharakis, I. Dimolitsas, D. Dechouniotis, S. Papavassiliou, COSMOS: An orchestration framework for smart computation offloading in edge clouds, in: NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium, IEEE, 2020, pp. 1–6.

[147] M. Akter, F.T. Zohra, A.K. Das, Q-MAC: Qos and mobility aware optimal resource allocation for dynamic application offloading in mobile cloud computing, in: 2017 International Conference on Electrical, Computer and Communication Engineering, ECCE, IEEE, 2017, pp. 803–808.

[148] C. Puliafito, E. Mingozzi, C. Vallati, F. Longo, G. Merlino, Companion fog computing: Supporting things mobility through container migration at the edge, in: 2018 IEEE International Conference on Smart Computing, SMARTCOMP, 2018, pp. 97–105.

[149] W. Junior, A. França, K. Dias, J.N. de Souza, Supporting mobility-aware computational offloading in mobile cloud environment, J. Netw. Comput. Appl. 94 (2017) 93–108.

[150] L. Yang, J. Cao, H. Cheng, Y. Ji, Multi-user computation partitioning for latency sensitive mobile cloud applications, IEEE Trans. Comput. 64 (8) (2014) 2253–2266.

[151] S. Bi, Y.J. Zhang, Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading, IEEE Trans. Wireless Commun. 17 (6) (2018) 4177–4190.

[152] T. Zhao, S. Zhou, X. Guo, Z. Niu, Tasks scheduling and resource allocation in heterogeneous cloud for delay-bounded mobile edge computing, in: 2017 IEEE International Conference on Communications, ICC, IEEE, 2017, pp. 1–7.

[153] Q. Liu, T. Han, N. Ansari, Joint radio and computation resource management for low latency mobile edge computing, in: 2018 IEEE Global Communications Conference, GLOBECOM, IEEE, 2018, pp. 1–7.

[154] L. Yang, J. Cao, Y. Yuan, T. Li, A. Han, A. Chan, A framework for partitioning and execution of data stream applications in mobile cloud computing, ACM SIGMETRICS Perform. Eval. Rev. 40 (4) (2013) 23–32.

[155] J. Liu, Y. Mao, J. Zhang, K.B. Letaief, Delay-optimal computation task scheduling for mobile-edge computing systems, in: 2016 IEEE International Symposium on Information Theory, ISIT, IEEE, 2016, pp. 1451–1455.

[156] M. Jia, W. Liang, Z. Xu, M. Huang, Y. Ma, Qos-aware cloudlet load balancing in wireless metropolitan area networks, IEEE Trans. Cloud Comput. (2018).

[157] C. Wang, C. Liang, F.R. Yu, Q. Chen, L. Tang, Computation offloading and resource allocation in wireless cellular networks with mobile edge computing, IEEE Trans. Wireless Commun. 16 (8) (2017) 4924–4938.

[158] C. Sonmez, A. Ozgovde, C. Ersoy, Fuzzy workload orchestration for edge computing, IEEE Trans. Netw. Serv. Manag. (2019).

[159] C.-F. Liu, M. Bennis, H.V. Poor, Latency and reliability-aware task offloading and resource allocation for mobile edge computing, in: 2017 IEEE Globecom Workshops, GC Wkshps, IEEE, 2017, pp. 1–7.

[160] L. Wang, L. Jiao, J. Li, M. Mühlhäuser, Online resource allocation for arbitrary user mobility in distributed edge clouds, in: 2017 IEEE 37th International Conference on Distributed Computing Systems, ICDCS, IEEE, 2017, pp. 1281–1290.

[161] S. Cao, X. Tao, Y. Hou, Q. Cui, An energy-optimal offloading algorithm of mobile computing based on hetnets, in: 2015 International Conference on Connected Vehicles and Expo, ICCVE, IEEE, 2015, pp. 254–258.

[162] Y. Zhao, S. Zhou, T. Zhao, Z. Niu, Energy-efficient task offloading for multiuser mobile cloud computing, in: 2015 IEEE/CIC International Conference on Communications in China, ICCC, IEEE, 2015, pp. 1–5.

[163] Y. Wang, M. Sheng, X. Wang, L. Wang, W. Han, Y. Zhang, Y. Shi, Energy-optimal partial computation offloading using dynamic voltage scaling, in: 2015 IEEE International Conference on Communication Workshop, ICCW, IEEE, 2015, pp. 2695–2700.

[164] C. You, K. Huang, Multiuser resource allocation for mobile-edge computation offloading, in: 2016 IEEE Global Communications Conference, GLOBECOM, IEEE, 2016, pp. 1–6.

[165] P. Di Lorenzo, S. Barbarossa, S. Sardellitti, Joint optimization of radio resources and code partitioning in mobile edge computing, 2013, arXiv preprint arXiv:1307.3835.

[166] W. Labidi, M. Sarkiss, M. Kamoun, Energy-optimal resource scheduling and computation offloading in small cell networks, in: 2015 22nd International Conference on Telecommunications, ICT, IEEE, 2015, pp. 313–318.

[167] M. Kamoun, W. Labidi, M. Sarkiss, Joint resource allocation and offloading strategies in cloud enabled cellular networks, in: 2015 IEEE International Conference on Communications, ICC, IEEE, 2015, pp. 5529–5534.

[168] S. Sardellitti, G. Scutari, S. Barbarossa, Joint optimization of radio and computational resources for multicell mobile-edge computing, IEEE Trans. Signal Inf. Process. Netw. 1 (2) (2015) 89–103.

[169] W. Labidi, M. Sarkiss, M. Kamoun, Joint multi-user resource scheduling and computation offloading in small cell networks, in: 2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob, IEEE, 2015, pp. 794–801.

[170] Y. Mao, J. Zhang, K.B. Letaief, Dynamic computation offloading for mobile-edge computing with energy harvesting devices, IEEE J. Sel. Areas Commun. 34 (12) (2016) 3590–3605.

[171] K. Singh, A. Awasthi, Quality, Reliability, Security and Robustness in Heterogeneous Networks, Springer, 2013.

[172] A. Kiani, N. Ansari, Optimal code partitioning over time and hierarchical cloudlets, IEEE Commun. Lett. 22 (1) (2017) 181–184.

[173] Y. Mao, J. Zhang, K.B. Letaief, Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems, in: 2017 IEEE Wireless Communications and Networking Conference, WCNC, IEEE, 2017, pp. 1–6.

[174] J. Zhang, X. Hu, Z. Ning, E.C.-H. Ngai, L. Zhou, J. Wei, J. Cheng, B. Hu, Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks, IEEE Internet Things J. 5 (4) (2017) 2633–2645.

[175] J. Guo, Z. Song, Y. Cui, Z. Liu, Y. Ji, Energy-efficient resource allocation for multi-user mobile edge computing, in: GLOBECOM 2017-2017 IEEE Global Communications Conference, IEEE, 2017, pp. 1–7.

[176] F. Samie, V. Tsoutsouras, L. Bauer, S. Xydis, D. Soudris, J. Henkel, Computation offloading and resource allocation for low-power IoT edge devices, in: 2016 IEEE 3rd World Forum on Internet of Things, WF-IoT, IEEE, 2016, pp. 7–12.

[177] A. Leivadeas, C. Papagianni, S. Papavassiliou, Going Green with the Networked Cloud: Methodologies and Assessment, John Wiley & Sons, Inc., 2015.

[178] S. Li, N. Zhang, S. Lin, L. Kong, A. Katangur, M.K. Khan, M. Ni, G. Zhu, Joint admission control and resource allocation in edge computing for Internet of Things, IEEE Netw. 32 (1) (2018) 72–79.

[179] L. Tong, Y. Li, W. Gao, A hierarchical edge cloud architecture for mobile computing, in: IEEE INFOCOM 2016-the 35th Annual IEEE International Conference on Computer Communications, IEEE, 2016, pp. 1–9.

[180] A. Leivadeas, G. Kesidis, M. Ibnkahla, I. Lambadaris, VNF placement optimization at the edge and cloud, Future Internet 11 (3) (2019) 69.

[181] A. Leivadeas, G. Kesidis, M. Falkner, I. Lambadaris, A graph partitioning game theoretical approach for the VNF service chaining problem, IEEE Trans. Netw. Serv. Manag. 14 (4) (2017) 890–903.

[182] A. Leivadeas, M. Falkner, I. Lambadaris, M. Ibnkahla, G. Kesidis, Balancing delay and cost in virtual network function placement and chaining, in: 2018 IEEE International Conference on Network Softwarization and Workshops, NetSoft, IEEE, 2018, pp. 433–440.

[183] I. Skarga-Bandurova, M. Derkach, I. Kotsiuba, The information service for delivering arrival public transport prediction, in: 2018 IEEE 4th International Symposium on Wireless Systems Within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems, IDAACS-SWS, 2018, pp. 191–195.

[184] A. Hameed, A. Leivadeas, IoT Traffic multi-classification using network and statistical features in a smart environment, in: EEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, CAMAD, IEEE, 2020, pp. 1–7.

[185] W. Sun, J. Liu, Y. Yue, AI-enhanced offloading in edge computing: when machine learning meets industrial IoT, IEEE Netw. 33 (5) (2019) 68–74.

[186] A. Botchkarev, A new typology design of performance metrics to measure errors in machine learning regression algorithms, Interdisciplinary J. Inf. Knowl. Manag. 14 (2019) 045–076.

[187] J.-O. Palacio-Niño, F. Berzal, Evaluation metrics for unsupervised learning algorithms, 2019, arXiv preprint arXiv:1905.05667.

[188] J. Lever, M. Krzywinski, N. Altman, Classification evaluation, Nature Methods 13 (8) (2016) 603–604, https://www.nature.com/articles/nmeth.3945.

[189] Y. Wang, M. Sheng, X. Wang, L. Wang, J. Li, Mobile-edge computing: Partial computation offloading using dynamic voltage scaling, IEEE Trans. Commun. 64 (10) (2016) 4268–4282.

[190] M. Deng, H. Tian, B. Fan, Fine-granularity based application offloading policy in cloud-enhanced small cell networks, in: 2016 IEEE International Conference on Communications Workshops, ICC, IEEE, 2016, pp. 638–643.

[191] Y. Mao, J. Zhang, S. Song, K.B. Letaief, Power-delay tradeoff in multi-user mobile-edge computing systems, in: 2016 IEEE Global Communications Conference, GLOBECOM, IEEE, 2016, pp. 1–6.

[192] C. You, K. Huang, H. Chae, B.-H. Kim, Energy-efficient resource allocation for mobile-edge computation offloading, IEEE Trans. Wireless Commun. 16 (3) (2016) 1397–1411.
[193] M.-H. Chen, B. Liang, M. Dong, Joint offloading and resource allocation for computation and communication in mobile cloud with computing access point, in: IEEE INFOCOM 2017-IEEE Conference on Computer Communications, IEEE, 2017, pp. 1–9.
[194] M.-H. Chen, B. Liang, M. Dong, A semidefinite relaxation approach to mobile cloud offloading with computing access point, in: 2015 IEEE 16th International Workshop on Signal Processing Advances in Wireless Communications, SPAWC, IEEE, 2015, pp. 186–190.
[195] Y. Liu, F.R. Yu, X. Li, H. Ji, V.C. Leung, Hybrid computation offloading in fog and cloud networks with non-orthogonal multiple access, in: IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS, IEEE, 2018, pp. 154–159.
[196] Q.-V. Pham, T. Leanh, N.H. Tran, B.J. Park, C.S. Hong, Decentralized computation offloading and resource allocation for mobile-edge computing: A matching game approach, IEEE Access 6 (2018) 75868–75885.
[197] Z. Ning, P. Dong, X. Kong, F. Xia, A cooperative partial computation offloading scheme for mobile edge computing enabled Internet of Things, IEEE Internet Things J. (2018).
[198] J. Du, L. Zhao, J. Feng, X. Chu, Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee, IEEE Trans. Commun. 66 (4) (2018) 1594–1608.
[199] S. Li, Y. Tao, X. Qin, L. Liu, Z. Zhang, P. Zhang, Energy-aware mobile edge computation offloading for IoT over heterogenous networks, IEEE Access 7 (2019) 13092–13105.
[200] H. Guo, J. Liu, J. Zhang, Computation offloading for multi-access mobile edge computing in ultra-dense networks, IEEE Commun. Mag. 56 (8) (2018) 14–19.
[201] X. Chen, L. Jiao, W. Li, X. Fu, Efficient multi-user computation offloading for mobile-edge cloud computing, IEEE/ACM Trans. Netw. 24 (5) (2015) 2795–2808.
[202] Y. Liu, C. Xu, Y. Zhan, Z. Liu, J. Guan, H. Zhang, Incentive mechanism for computation offloading using edge computing: A Stackelberg game approach, Comput. Netw. 129 (2017) 399–409.
[203] H. Guo, J. Liu, Collaborative computation offloading for multiaccess edge computing over fiber–wireless networks, IEEE Trans. Veh. Technol. 67 (5) (2018) 4514–4526.
[204] M. Zeng, Y. Li, K. Zhang, M. Waqas, D. Jin, Incentive mechanism design for computation offloading in heterogeneous fog computing: A contract-based approach, in: 2018 IEEE International Conference on Communications, ICC, 2018, pp. 1–6.
[205] J. Du, E. Gelenbe, C. Jiang, H. Zhang, Y. Ren, Contract design for traffic offloading and resource allocation in heterogeneous ultra-dense networks, IEEE J. Sel. Areas Commun. 35 (11) (2017) 2457–2467.
[206] Z. Hu, Z. Zheng, L. Song, T. Wang, X. Li, UAV offloading: Spectrum trading contract design for UAV-assisted cellular networks, IEEE Trans. Wireless Commun. 17 (9) (2018) 6093–6107.
[207] Y. Li, J. Zhang, X. Gan, L. Fu, H. Yu, X. Wang, A contract-based incentive mechanism for delayed traffic offloading in cellular networks, IEEE Trans. Wireless Commun. 15 (8) (2016) 5314–5327.
[208] M.G.C. Resende, P. Pardalos, Handbook of Optimization in Telecommunication, Springer, 2006.
[209] D. Monderer, L.S. Shapley, Potential games, Games Economic Behav. 14 (1) (1996) 124–143.
[210] Y. Zhang, M. Pan, L. Song, Z. Dawy, Z. Han, A survey of contract theory-based incentive mechanism design in wireless networks, IEEE Wirel. Commun. 24 (3) (2017) 80–85.
[211] M. Gendreau, J.Y. Potvin, Handbook of Metaheuristics, Int. Series in Operations Research & Management Science, 2010.
[212] K.B. Letaief, W. Chen, Y. Shi, J. Zhang, Y.-J.A. Zhang, The roadmap to 6G: AI empowered wireless networks, IEEE Commun. Mag. 57 (8) (2019) 84–90.
[213] F.A. Salaht, F. Desprez, A. Lebre, C. Prud'Homme, M. Abderrahim, Service placement in fog computing using constraint programming, in: 2019 IEEE International Conference on Services Computing, SCC, IEEE, 2019, pp. 19–27.
[214] A.I. Orhean, F. Pop, I. Raicu, New scheduling approach using reinforcement learning for heterogeneous distributed systems, J. Parallel Distrib. Comput. 117 (2018) 292–302.
[215] Y.-R. Shiue, K.-C. Lee, C.-T. Su, Real-time scheduling for a smart factory using a reinforcement learning approach, Comput. Ind. Eng. 125 (2018) 604–614.
[216] S. Wang, T. Tuor, T. Salonidis, K.K. Leung, C. Makaya, T. He, K. Chan, Adaptive federated learning in resource constrained edge computing systems, IEEE J. Sel. Areas Commun. 37 (6) (2019) 1205–1221.
[217] L.C. Monteiro, F.C. Delicato, L. Pirmez, P.F. Pires, C. Miceli, Dpcas: Data prediction with cubic adaptive sampling for wireless sensor networks, in: International Conference on Green, Pervasive, and Cloud Computing, Springer, 2017, pp. 353–368.
[218] B. Yang, X. Cao, J. Bassey, X. Li, T. Kroecker, L. Qian, Computation offloading in multi-access edge computing networks: A multi-task learning approach, in: ICC 2019 - 2019 IEEE International Conference on Communications, ICC, 2019, pp. 1–6 (ISSN: 1938-1883).
[219] D. Rahbari, M. Nickray, Task offloading in mobile fog computing by classification and regression tree, Peer-to-Peer Netw. Appl. 13 (1) (2020) 104–122.
[220] W. Junior, E. Oliveira, A. Santos, K. Dias, A context-sensitive offloading system using machine-learning classification algorithms for mobile cloud environment, Future Gener. Comput. Syst. 90 (2019) 503–520.
[221] H. Bashir, S. Lee, K.H. Kim, Resource allocation through logistic regression and multicriteria decision making method in IoT fog computing, Trans. Emerg. Telecommun. Technol. n/a (n/a) (2019) e3824, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/ett.3824, https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.3824.
[222] Y. Jararweh, M.B. Issa, M. Daraghmeh, M. Al-Ayyoub, M.A. Alsmirat, Energy efficient dynamic resource management in cloud computing based on logistic regression model and median absolute deviation, Sustain. Comput.: Inform. Syst. 19 (2018) 262–274.
[223] J.C. Guevara, R.d.S. Torres, N.L.S. da Fonseca, On the classification of fog computing applications: A machine learning perspective, J. Netw. Comput. Appl. 159 (2020) 102596.
[224] L. Liu, D. Qi, N. Zhou, Y. Wu, A task scheduling algorithm based on classification mining in fog computing environment, Wirel. Commun. Mob. Comput. (ISSN: 1530-8669) 2018 (2018) e2102348, https://www.hindawi.com/journals/wcmc/2018/2102348/.
[225] K. Kim, J. Lynskey, S. Kang, C.S. Hong, Prediction based sub-task offloading in mobile edge computing, in: 2019 International Conference on Information Networking, ICOIN, 2019, pp. 448–452 (ISSN: 1976-7684).
[226] F. Farahnakian, P. Liljeberg, J. Plosila, LiRCUP: Linear regression based cpu usage prediction algorithm for live migration of virtual machines in data centers, in: 2013 39th Euromicro Conference on Software Engineering and Advanced Applications, 2013, pp. 357–364 (ISSN: 2376-9505).
[227] R.A.C. da Silva, N.L.S.d. Fonseca, Resource allocation mechanism for a fog-cloud infrastructure, in: 2018 IEEE International Conference on Communications, ICC, 2018, pp. 1–6 (ISSN: 1938-1883).
[228] R. Hu, J. Jiang, G. Liu, L. Wang, CPU load prediction using support vector regression and Kalman Smoother for Cloud, in: 2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops, 2013, pp. 88–92 (ISSN: 2332-5666).
[229] F. Farahnakian, T. Pahikkala, P. Liljeberg, J. Plosila, Energy aware consolidation algorithm based on k-nearest neighbor regression for cloud data centers, in: 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing, 2013, pp. 256–259.
[230] H. Cheng, W. Xia, F. Yan, L. Shen, Balanced clustering and joint resources allocation in cooperative fog computing system, in: 2019 IEEE Global Communications Conference, GLOBECOM, 2019, pp. 1–6 (ISSN: 2576-6813).
[231] M. Bouet, V. Conan, Mobile edge computing resources optimization: A geo-clustering approach, IEEE Trans. Netw. Serv. Manag. 15 (2) (2018) 787–796.
[232] Y. Li, N.T. Anh, A.S. Nooh, K. Ra, M. Jo, Dynamic mobile cloudlet clustering for fog computing, in; 2018 International Conference on Electronics, Information, and Communication, ICEIC, 2018, pp. 1–4.
[233] G. Li, Q. Lin, J. Wu, Y. Zhang, J. Yan, Dynamic computation offloading based on graph partitioning in mobile edge computing, IEEE Access 7 (2019) 185131–185139.
[234] L. Hu, Y. Tian, J. Yang, T. Taleb, L. Xiang, Y. Hao, Ready player one: UAV-clustering-based multi-task offloading for vehicular VR/AR gaming, IEEE Netw. 33 (3) (2019) 42–48.
[235] X. Liu, J. Yu, J. Wang, Y. Gao, Resource allocation with edge computing in IoT networks via machine learning, IEEE Internet Things J. 7 (4) (2020) 3415–3426.
[236] A.A.A. Gad-Elrab, A.Y. Noaman, Fuzzy clustering-based task allocation approach using bipartite graph in cloud-fog environment, in: Proceedings of the 16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, MobiQuitous '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 454–463, https://doi.org/10.1145/3360774.3360833.
[237] I. Ullah, H.Y. Youn, Task classification and scheduling based on K-means clustering for edge computing, Wirel. Pers. Commun. 113 (4) (2020) 2611–2624, https://doi.org/10.1007/s11277-020-07343-w.
[238] A. Bozorgchenani, D. Tarchi, G.E. Corazza, An energy-aware offloading clustering approach (eaoca) in fog computing, in: 2017 International Symposium on Wireless Communication Systems, ISWCS, 2017, pp. 390–395 (ISSN: 2154-0225).
[239] S. Yu, X. Wang, R. Langar, Computation offloading for mobile edge computing: A deep learning approach, in: 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications, PIMRC, 2017, pp. 1–6 (ISSN: 2166-9589).
[240] D.S. Rani, M. Pounambal, Deep learning based dynamic task offloading in mobile cloudlet environments, Evol. Intell. (ISSN: 1864-5917) (2019) https://doi.org/10.1007/s12065-019-00284-9.
[241] L. Huang, X. Feng, A. Feng, Y. Huang, L.P. Qian, Distributed deep learning-based offloading for mobile edge computing networks, Mob. Netw. Appl. (2018) https://doi.org/10.1007/s11036-018-1177-x.

[242] X. Wang, X. Wei, L. Wang, A deep learning based energy-efficient computational offloading method in internet of vehicles, China Commun. 16 (3) (2019) 81–91.
[243] Z. Ali, L. Jiao, T. Baker, G. Abbas, Z.H. Abbas, S. Khaf, A deep learning approach for energy efficient computational offloading in mobile edge computing, IEEE Access 7 (2019) 149623–149633.
[244] X. Zhu, S. Chen, S. Chen, G. Yang, Energy and delay co-aware computation offloading with deep learning in fog computing networks, in: 2019 IEEE 38th International Performance Computing and Communications Conference, IPCCC, 2019, pp. 1–6 (ISSN: 2374-9628).
[245] L. Huang, S. Bi, Y.J. Zhang, Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks, IEEE Trans. Mob. Comput. (2019) 1.
[246] R. Zhao, X. Wang, J. Xia, L. Fan, Deep reinforcement learning based mobile edge computing for intelligent Internet of Things, Phys. Commun. (2020) 101184.
[247] K. Zhang, Y. Zhu, S. Leng, Y. He, S. Maharjan, Y. Zhang, Deep learning empowered task offloading for mobile edge computing in urban informatics, IEEE Internet Things J. 6 (5) (2019) 7635–7647.
[248] H. Lu, C. Gu, F. Luo, W. Ding, X. Liu, Optimization of lightweight task offloading strategy for mobile edge computing based on deep reinforcement learning, Future Gener. Comput. Syst. 102 (2020) 847–861, http://www.sciencedirect.com/science/article/pii/S0167739X19308209.
[249] J. Baek, G. Kaddoum, Heterogeneous task offloading and resource allocations via deep recurrent reinforcement learning in partial observable multi-fog networks, IEEE Internet Things J. (2020) 1.
[250] J. Liu, X. Wei, T. Wang, J. Wang, An ant colony optimization fuzzy clustering task scheduling algorithm in mobile edge computing, in: J. Li, Z. Liu, H. Peng (Eds.), Security and Privacy in New Computing Environments, in: Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Springer International Publishing, Cham, 2019, pp. 615–624.
[251] M.K. Hussein, M.H. Mousa, Efficient task offloading for IoT-based applications in fog computing using ant colony optimization, IEEE Access 8 (2020) 37191–37201.
[252] T.G. Rodrigues, K. Suto, H. Nishiyama, N. Kato, A PSO model with VM migration and transmission power control for low service delay in the multiple cloudlets ECC scenario, in: 2017 IEEE International Conference on Communications, ICC, IEEE, 2017, pp. 1–6.
[253] L.N.T. Huynh, Q.-V. Pham, X.-Q. Pham, T.D.T. Nguyen, M.D. Hossain, E.-N. Huh, Efficient computation offloading in multi-tier multi-access edge computing systems: A particle swarm optimization approach, Appl. Sci. 10 (1) (2020) 203, https://www.mdpi.com/2076-3417/10/1/203.
[254] Y. Zhang, Y. Liu, J. Zhou, J. Sun, K. Li, Slow-movement particle swarm optimization algorithms for scheduling security-critical tasks in resource-limited mobile edge computing, Future Gener. Comput. Syst. 112 (2020) 148–161, http://www.sciencedirect.com/science/article/pii/S0167739X19333552.
[255] D. Zhang, F. Haider, M. St-Hilaire, C. Makaya, Model and algorithms for the planning of fog computing networks, IEEE Internet Things J. 6 (2) (2019) 3873–3884.
[256] A.A. Al-habob, O.A. Dobre, A. Garcia Armada, Sequential task scheduling for mobile edge computing using genetic algorithm, in: 2019 IEEE Globecom Workshops, GC Wkshps, 2019, pp. 1–6.
[257] V. Yadav, B.V. Natesha, R.M.R. Guddeti, GA-PSO: Service allocation in fog computing environment using hybrid bio-inspired algorithm, in: TENCON 2019 - 2019 IEEE Region 10 Conference, TENCON, 2019, pp. 1280–1285 (ISSN: 2159-3450).
[258] Z. Wang, Z. Zhao, G. Min, X. Huang, Q. Ni, R. Wang, User mobility aware task assignment for mobile edge computing, Future Gener. Comput. Syst. 85 (2018) 1–8, http://www.sciencedirect.com/science/article/pii/S0167739X17318587.
[259] M.B. Kamal, N. Javaid, S.A.A. Naqvi, H. Butt, T. Saif, M.D. Kamal, Heuristic min-conflicts optimizing technique for load balancing on fog computing, in: Advances in Intelligent Networking and Collaborative Systems, 2019, pp. 207–219.
[260] G. Amarasinghe, M.D. de Assun¸c ao, A. Harwood, S. Karunasekera, A data stream processing optimisation framework for edge computing applications, in: 2018 IEEE 21st International Symposium on Real-Time Distributed Computing, ISORC, 2018, pp. 91–98 (ISSN: 2375-5261).
[261] F. Ait Salaht, F. Desprez, A. Lebre, C. Prud'homme, M. Abderrahim, Service placement in fog computing using constraint programming, in: 2019 IEEE International Conference on Services Computing, SCC, 2019, pp. 19–27 (ISSN: 2474-2473).
[262] S. Vakilinia, D. Qiu, M.M. Ali, Optimal multi-dimensional dynamic resource allocation in mobile cloud computing, EURASIP J. Wireless Commun. Networking 2014 (1) (2014) 201, https://doi.org/10.1186/1687-1499-2014-201.
[263] M.C. Calzarossa, L. Massari, D. Tessera, Workload characterization: A survey revisited, ACM Comput. Surv. 48 (3) (2016) 48:1–48:43, https://doi.org/10.1145/2856127.
[264] I. Bouras, F. Aisopos, J. Violos, G. Kousiouris, A. Psychas, T. Varvarigou, G. Xydas, D. Charilas, Y. Stavroulas, Mapping of quality of service requirements to resource demands for iaas, in: CLOSER, 2019.
[265] C. Lecoutre, Constraint Networks: Targeting Simplicity for Techniques and Algorithms, John Wiley & Sons, 2013.
[266] F. Rossi, P. Van Beek, T. Walsh, Handbook of Constraint Programming, Elsevier, 2006.
[267] X. Yin, A. Jindal, V. Sekar, B. Sinopoli, A control-theoretic approach for dynamic adaptive video streaming over HTTP, in: Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, 2015, pp. 325–338.
[268] D.Y. Zhang, D. Wang, An integrated top-down and bottom-up task allocation approach in social sensing based edge computing systems, in: IEEE INFOCOM 2019-IEEE Conference on Computer Communications, IEEE, 2019, pp. 766–774.
[269] C. Wu, Y. Zhang, Y. Deng, Toward fast and distributed computation migration system for edge computing in IoT, IEEE Internet Things J. 6 (6) (2019) 10041–10052.
[270] Y. Li, Edge Computing-Based Access Network Selection for Heterogeneous Wireless Networks (Ph.D. dissertation), Universite Rennes, 2017.
[271] A. Koike, N. Higo, Y. Sueda, Proxy-based network function to assist robotic feedback control system, in: 2018 IEEE International Symposium on Local and Metropolitan Area Networks, LANMAN, IEEE, 2018, pp. 116–118.
[272] S. Host, W. Tärneberg, P. Ödling, M. Kihl, M. Savi, M. Tornatore, Network requirements for latency-critical services in a full cloud deployment, in: 2016 24th International Conference on Software, Telecommunications and Computer Networks, SoftCOM, IEEE, 2016, pp. 1–5.
[273] M. Maheswaran, T. Yang, S. Memon, A fog computing framework for autonomous driving assist: architecture, experiments, and challenges, 2019, arXiv preprint arXiv:1907.09454.
[274] N. Kalatzis, M. Avgeris, D. Dechouniotis, K. Papadakis-Vlachopapadopoulos, I. Roussaki, S. Papavassiliou, Edge computing in IoT ecosystems for UAV-enabled early fire detection, in: 2018 IEEE International Conference on Smart Computing, SMARTCOMP, IEEE, 2018, pp. 106–114.
[275] D. Spatharakis, M. Avgeris, N. Athanasopoulos, D. Dechouniotis, S. Papavassiliou, A switching offloading mechanism for path planning and localization in robotic applications, in: 2020 International Conferences on Internet of Things (IThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics), IEEE, 2020, pp. 77–84.
[276] M. Avgeris, D. Dechouniotis, N. Athanasopoulos, S. Papavassiliou, Adaptive resource allocation for computation offloading: A control-theoretic approach, ACM Trans. Internet Technol. (TOIT) 19 (2) (2019) 1–20.
[277] T. Dlamini, Á.F. Gambín, D. Munaretto, M. Rossi, Online resource management in energy harvesting BS sites through prediction and soft-scaling of computing resources, in: 2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC, IEEE, 2018, pp. 1820–1826.
[278] L. Zhang, S. Wang, R.N. Chang, QCSS: a QoE-aware control plane for adaptive streaming service over mobile edge computing infrastructures, in: 2018 IEEE International Conference on Web Services, ICWS, IEEE, 2018, pp. 139–146.
[279] F. Basic, A. Aral, I. Brandic, Fuzzy handoff control in edge offloading, in: 2019 IEEE International Conference on Fog Computing, ICFC, IEEE, 2019, pp. 87–96.
[280] P. Skarin, J. Eker, M. Kihl, K. Årzén, Cloud-assisted model predictive control, in: 2019 IEEE International Conference on Edge Computing, EDGE, 2019, pp. 110–112.
[281] R.E. Kalman, et al., Contributions to the theory of optimal control, Bol. Soc. Mat. Mexicana 5 (2) (1960) 102–119.
[282] Z. Zhou, J. Yu, X. Dong, Q. Li, Z. Ren, Robust adaptive attitude control of the quad-rotor UAV based on the LQR and NESO technique, in: 2018 IEEE 14th International Conference on Control and Automation, ICCA, IEEE, 2018, pp. 745–750.
[283] A. Ullah, J. Li, Y. Shen, A. Hussain, A control theoretical view of cloud elasticity: taxonomy, survey and challenges, Cluster Comput. 21 (4) (2018) 1735–1764.
[284] D. Dechouniotis, N. Leontiou, N. Athanasopoulos, G. Bitsoris, S. Denazis, ACRA: A unified admission control and resource allocation framework for virtualized environments, in: 2012 8th International Conference on Network and Service Management (Cnsm) and 2012 Workshop on Systems Virtualiztion Management (Svm), IEEE, 2012, pp. 145–149.
[285] P.S. Saikrishna, R. Pasumarthy, N.P. Bhatt, Identification and multivariable gain-scheduling control for cloud computing systems, IEEE Trans. Control Syst. Technol. 25 (3) (2016) 792–807.
[286] D. Dechouniotis, N. Leontiou, N. Athanasopoulos, A. Christakidis, S. Denazis, A control-theoretic approach towards joint admission control and resource allocation of cloud computing services, Int. J. Netw. Manag. 25 (3) (2015) 159–180.
[287] S. Rashidi, S. Sharifian, Cloudlet dynamic server selection policy for mobile task off-loading in mobile cloud computing using soft computing techniques, J. Supercomput. 73 (9) (2017) 3796–3820.
[288] N. Leontiou, D. Dechouniotis, S. Denazis, S. Papavassiliou, A hierarchical control framework of load balancing and resource allocation of cloud computing services, Comput. Electr. Eng. 67 (2018) 235–251.

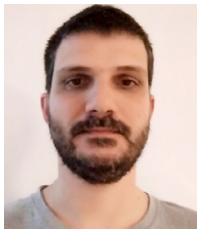
[289] D. Dechouniotis, N. Athanasopoulos, A. Leivadeas, N. Mitton, R. Jungers, S. Papavassiliou, Edge computing resource allocation for dynamic networks: The DRUID-net vision and perspective, Sensors 20 (8) (2020) 2191.
[290] W. Zhang, M.S. Branicky, S.M. Phillips, Stability of networked control systems, IEEE Control Syst. Mag. 21 (1) (2001) 84–99.
[291] J.P. Hespanha, P. Naghshtabrizi, Y. Xu, A survey of recent results in networked control systems, Proc. IEEE 95 (1) (2007) 138–162.
[292] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M.I. Jordan, S.S. Sastry, Kalman filtering with intermittent observations, IEEE Trans. Autom. Control 49 (9) (2004) 1453–1464.
[293] D. Simon, D. Robert, O. Sename, Robust control/scheduling co-design: application to robot control, in: 11th IEEE Real Time and Embedded Technology and Applications Symposium, IEEE, 2005, pp. 118–127.
[294] W. Heemels, K.H. Johansson, P. Tabuada, An introduction to event-triggered and self-triggered control, in: 2012 IEEE 51st IEEE Conference on Decision and Control, CDC, IEEE, 2012, pp. 3270–3285.

Firdose Saeik is currently pursuing a Ph.D. at ETS. Canada. He received the M.Sc. degree from VIT university, India, in 2010. He has more than 9 years of work experience with multiple roles as System Engineer, Junior Researcher, and Research Engineer. His research interests are in the field of Mobile Edge Computing, Quality of Experience, Next-generation mobile services and applications such as Virtual Reality (VR) and Augmented Reality (AR).

Marios Avgeris is currently a Ph.D student in the NETMODE Lab at the National Technical University of Athens (NTUA). He received his Diploma in Electrical & Computer Engineering (ECE) from NTUA, Greece, in 2016. His research interests are control theory, edge and cloud computing, IoT, semantic web technologies and network monitoring.

Dimitrios Spatharakis is currently a Ph.D. student in the NETMODE Lab at the National Technical University of Athens (NTUA). He received a Diploma in Electrical & Computer Engineering (ECE) from NTUA, Greece, in 2018. His research interests focus on IoT, cyber–physical systems, edge computing and cloud computing

Nina Santi is a Ph.D. student under the supervision of Nathalie Mitton in the Inria FUN team. Their focus is on small computing devices like electronic tags and sensor networks. She has received the M.Sc. degrees in Computer Science from University of Lille, France, in 2020.



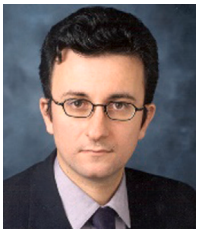
Dimitrios Dechouniotis is currently research associate with NETMODE Lab of the National Technical University of Athens (NTUA). From 2007 to 2016, he was non-tenured Lecturer at the EE Dept. of Technical Educational Institute of Western Greece, Greece. He received his diploma in ECE from University of Patras in 2004, the M.Sc. degree in Control Systems and Robotics from NTUA in 2009, and the Ph.D. degree in ECE from University of Patras in 2014. His research interests lie in the area of cloud computing, Internet of Things, mobile cloud computing and control theory.

John Violos is research associate in the Dept. of Software Engineering and Information Technology at ETS. His previous positions were research associate at National Technical University of Athens, sessional lecturer at Harokopio University of Athens and visiting lecturer at National and Kapodistrian University of Athens. He was a member in the European Commission's Digital Single Market working group on the code of conduct for switching and porting data between cloud service providers. His research interests include Deep Learning, Machine Learning, Cloud and Edge computing.

Aris Leivadeas is currently an Assistant Professor with the Dept. of Software and Information Technology Engineering at ETS. From 2015 to 2018 he was a postdoc in the Dept. of SCE, at Carleton University. In parallel, Aris worked as an intern at Ericsson and then at Cisco in Ottawa, Canada. He received his diploma in ECE from University of Patras in 2008, the M.Sc. degree in Engineering from King's College London in 2009, and the Ph.D degree in ECE from NTUA in 2015. His research interests include Cloud Computing, IoT, and network optimization and management. He received the best paper award in ICPE'18 and the best presentation award in HPSR'20.

Nikolaos Athanasopoulos is a Lecturer at the School of Electronics, Electrical Engineering and Computer Science at Queens University Belfast. He received a Diploma and a Ph.D. in Electrical and Computer Engineering from the University of Patras, Greece and has held postdoctoral researcher positions at TU/e and UCLouvain. He has been an IKY and a Marie Curie Fellow. His interests are in control theory, focusing on hybrid systems and set-based methods with applications in edge/cloud computing and resource allocation.

Nathalie Mitton received the M.Sc. and PhD. degrees in Computer Science from INSA Lyon in 2003 and 2006 respectively. She received her Habilitation à diriger des recherches (HDR) in 2011 from Université Lille 1. She is currently an Inria full researcher since 2006 and from 2012, she is the scientific head of the Inria FUN team which is focused on small computing devices like electronic tags and sensor networks. Her research interests focus on self-organization from PHY to routing for wireless constrained networks. She has published her research in more than 30 international revues and more than 100 international conferences. She is involved in the setup of the FIT IoT LAB platform (http://fit-equipex.fr/, https://www.iot-lab.info), the H2020 CyberSANE and VESSEDIA projects and in several program and organization committees such as Infocom 2021&2020&2019, PerCom 2020&2019, DCOSS 2021&2020&2019, Adhocnow (since 2015), ICC (since 2015), Globecom (since 2017), VTC (since 2016), etc. She also supervises several Ph.D. students and engineers.



Symeon Papavassiliou is currently a professor in the School of Electrical and Computer Engineering at the National Technical University of Athens (NTUA). From 1995 to 1999, he was a senior technical staff member at AT&T Laboratories, New Jersey. In August 1999 he joined the ECE Dept at the New Jersey Institute of Technology, USA, where he was an associate professor until 2004. He has an established record of publications in his field of expertise, with more than 350 technical journal and conference published papers, while he has received several scientific awards and distinctions. His main research interests lie in the areas of optimization and performance evaluation of mobile and distributed systems, wireless networks and complex systems.