LEBANESE AMERICAN UNIVERSITY

UAV-assisted Multi-tier Computing Framework for IoT Networks

By

ABEER ZIAD TOUT

A thesis

Submitted in partial fulfillment of the requirements

for the degree of Master of Science in Computer Science

School of Arts and Sciences

August 2022

LAU
الجامعة اللبنانية الأميركية
Lebanese American University

THESIS APPROVAL FORM

Student Name: Ms. Abeer Tout I.D. #: 202004564

Thesis Title: UAV-assisted Multi-tier Computing Framework for IoT Networks

Program: MS in Computer Science

Department: Computer Science and Mathematics

School: Arts and Sciences

The undersigned certify that they have examined the final electronic copy of this thesis and approved it in Partial Fulfillment of the requirements for the degree of:

Master of Science in the major of Computer Science

Thesis Advisor's [redacted]dine Abbas

Signature: [redacted] Date: 26 / 08 / 2022
Day Month Year

Committee Member's Name: Dr. Danielle Azar

Signature: [redacted] Date: 26 / 08 / 2022
Day Month Year

Committee Member's Name: Dr. Eileen-Marie Hanna

Signature: [redacted] Date: 26 / 08 / 2022
Day Month Year

THESIS COPYRIGHT RELEASE FORM

LEBANESE AMERICAN UNIVERSITY NON-EXCLUSIVE DISTRIBUTION LICENSE

By signing and submitting this license, you (the author(s) or copyright owner) grants the Lebanese American University (LAU) the non-exclusive right to reproduce, translate (as defined below), and/or distribute your submission (including the abstract) worldwide in print and electronic formats and in any medium, including but not limited to audio or video. You agree that LAU may, without changing the content, translate the submission to any medium or format for the purpose of preservation. You also agree that LAU may keep more than one copy of this submission for purposes of security, backup and preservation. You represent that the submission is your original work, and that you have the right to grant the rights contained in this license. You also represent that your submission does not, to the best of your knowledge, infringe upon anyone's copyright. If the submission contains material for which you do not hold copyright, you represent that you have obtained the unrestricted permission of the copyright owner to grant LAU the rights required by this license, and that such third-party owned material is clearly identified and acknowledged within the text or content of the submission. IF THE SUBMISSION IS BASED UPON WORK THAT HAS BEEN SPONSORED OR SUPPORTED BY AN AGENCY OR ORGANIZATION OTHER THAN LAU, YOU REPRESENT THAT YOU HAVE FULFILLED ANY RIGHT OF REVIEW OR OTHER OBLIGATIONS REQUIRED BY SUCH CONTRACT OR AGREEMENT. LAU will clearly identify your name(s) as the author(s) or owner(s) of the submission, and will not make any alteration, other than as allowed by this license, to your submission.

Name: Abeer Tout

Signature: [redacted]

Date: 25/06/2022

الجامعة اللبنانية الأميركية
Lebanese American University

PLAGIARISM POLICY COMPLIANCE STATEMENT

I certify that:

1. I have read and understood LAU's Plagiarism Policy.
2. I understand that failure to comply with this Policy can lead to academic and disciplinary actions against me.
3. This work is substantially my own, and to the extent that any part of this work is not my own I have indicated that by acknowledging its sources.

Name: Abeer Tout

Signature: [redacted]

Date: 25/06/2022

ACKNOWLEDGEMENTS

This project would not have been possible without the support of many people. Many thanks to my advisor, Dr. Sanaa Sharafeddine, and my Co-advisor, Dr. Nadine Abbas, who read my numerous revisions and helped make some sense of the confusion. Also thanks to my committee members, Dr. Danielle Azar, and Dr. Eileen-Marie Hanna, who offered guidance and support. And finally, thanks to my parents, and friends who endured this long process with me, always offering support and love.

UAV-assisted Multi-tier Computing Framework for IoT Networks

Abeer Tout

ABSTRACT

Next-generation wireless networks are expected to support a massive amount of Internet of Things (IoT) devices in delivering a wide range of novel resource-intensive applications. IoT devices, despite their enhanced capabilities, fall short of meeting the computational requirements of these applications within a strict deadline. This thesis leverages a hierarchical computational model where IoT devices offload their computational tasks to a number of unmanned aerial vehicles (UAVs) that can be deployed in a way to establish strong Line-of-Sight (LoS) links. Being equipped with computational resources, UAVs process part of the tasks within the set time threshold while the other tasks are transferred to edge and cloud servers for potential processing. This work aims at optimizing the number and position of deployed UAVs, IoT-to-UAV association, resource allocation, and task offloading to UAVs, edge servers, and the cloud, while ensuring various system constraints. The problem is formulated as a mixed integer programming problem and solved using Successive Convex Approximation. An efficient solution is then proposed to decompose the main problem into two subproblems that are solved iteratively. The first subproblem minimizes the number of IoT clusters and positions a UAV to serve each cluster while the second subproblem maximizes the percentage of admitted tasks using the resulting number of UAVs determined by the first subproblem. As long as not all devices are fully served, an additional UAV is introduced and the two subproblems are solved repeatedly. The proposed decomposition solution has been evaluated as a function of various system parameters and application use cases to show optimized performance and high scalability.

Keywords: Internet of Things (IoT), Unmanned Aerial Vehicle (UAV), Hierarchical computational model, Line-of-Sight (LoS), Successive Convex Approximation (SCA).

Table of Contents

List Of Figures

List Of Tables

Chapter 1

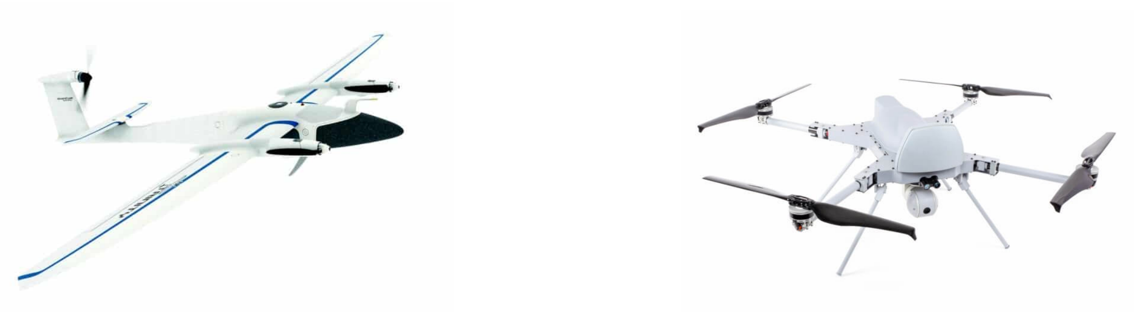
Introduction

1.1 Motivation and Background Information

While the deployment of fifth generation (5G) wireless networks is steadily rolling, major research efforts are underway to outline the road-map for the sixth generation networks (6G), as well as the developing trends, designs, and enabling technologies [1]. One of the most pressing 6G network requirements is to provide global coverage, connecting terrestrial networks with aerial and space-borne networks, in contrast to the 5G system, which keeps terrestrial networks as its main focus[2].

Considering the upcoming 6G networks, the use of UAVs as areal stations has been suggested to help locate terrestrial IoT sensors and offer relay services. [3].

UAVs, usually referred to drones or remotely piloted aircraft, have been more popular over the past couple decades due to their great mobility and low cost [4]. UAVs have traditionally been utilized primarily in hostile areas by the military to prevent pilot casualties. Small UAVs are now more widely available to the general public thanks to ongoing cost reductions and device miniaturization, which has led to the emergence of numerous new applications in the civilian and commercial sectors. Typical examples include climate tracking, forest fire identification, traffic monitoring, disaster search and rescue, relaying data and communication, etc. [5]. UAVs may be roughly divided into two groups: fixed wing and rotary wing (Figure



(a) Fixed Wing. (b) Rotary Wing.

Figure 1.1: Types of UAVs.

1.1), each having distinct advantages and disadvantages. For instance, fixed-wing UAVs frequently travel at high speeds and carry large payloads, but since they must continuously move forward to stay in the air, they are unsuitable for fixed functions like close examination. On the other hand, rotary-wing UAVs can fly in any direction and can remain fixed in the air despite having limited mobility and payload. As a result, the applications have a significant impact on the UAV selection.

Numerous applications of remote sensing have profited from the usage of unmanned aerial vehicles. Most of the time, this was due to the mission's expense, the requirement for immediate action or the requirement to undertake observations in a situation that may be unpleasant or risky to an aircrew. Figure 1.2 represents some applications of UAVs [6] [7].

In additions, UAV-aided wireless communication is a potential and essential part of future wireless systems, which must serve more operations while having higher instructions capacity than the systems currently used[7]. Here are three examples of common UAV-assisted wireless communications use cases:

-UAV-aided ubiquitous coverage refers to the situation in which UAVs are used to support the serving area's existing communications infrastructure.

-UAV-assisted relaying which is a technique where UAVs are used to offer wireless connectivity between two or more remote users who do not have strong and direct communication links.

-UAV-aided data collection and information dissemination, in which UAVs are sent

Figure 1.2: Applications of UAVs.

out to gather delay-tolerant data to a lot of dispersed wireless devices.

The size of UAVs varies widely, with the smallest weighing less than 1 kg and fitting easily into personal baggage, and the largest UAVs being the size of human aircraft. The size variation also relates to various norms and constraints. Currently, the FAA (Federal Aviation Administration) and the EASA (European Aviation Safety Agency) only allow UAVs with a take-off weight of less than 25 kg to fly at altitudes below 120 m, which correlates to uncontrolled airspace that manned aircraft are not permitted to fly in. In contrast, bigger UAVs are obliged to utilize regulated airspace and communicate with air traffic control.

For energy sources, in [8], they investigated the potential of deploying a network of UAV charging stations on city rooftops where the UAV will be replaced by another one while it is charging by the station. Another solution they looked into was battery hot-swapping, which involves using a docking station with a mechanical actuator to replace a UAV's exhausted battery with a new one. One other solution was to the laser power but it might not be appropriate for low-altitude UAVs in cities due to the presence of line-of-sight-blocking structures. Finally, they looked into the advancements in battery capacity. New innovations on the horizon promise to increase UAV flight time, alleviating the battery lifespan constraint of UAV infrastructure.

In [9], they tested the solar power as an energy source and, thus, allows small size UAVs to carry larger, more capable sensor payloads and can increase flight periods to more than 24 hours, allowing for multi-day flying .

In addition, for weather conditions, in [10], they used machine learning to have a continuous proactive situational awareness to improve operational safety. Numerous UAV activities occur at low heights, where the climate may be more complex than at higher heights. Additionally, small UAVs are highly sensitive to significant changes in climatic conditions due to their small size and light weight. Several elements make up weather awareness, including continuous recognition of high winds, stormy weather, and climatic zones. This information is helpful for anticipating the best air spaces and times before these climate challenges arise. The researchers in [11] estimated the 3D wind flow in less than two minutes using a deep convolutional neural network (CNN) to allow safer UAV routes in heavy wind situations. Reference [12] states that a secure UAV flight evaluation should be accomplished in less than 30 seconds.

Recently, the paradigm of UAV-assisted localization has begun to take place in 6G research, where UAVs act as areal stations and serve for locating ground-based devices [13]. In addition, due to their great mobility and low cost, UAVs are one of the finest choices for supplying airborne networks. In contrast to terrestrial base stations, which are usually characterized by a lack of line-of-sight (LoS), UAVs functioning as aerial stations are projected to significantly improve IoT networks by continuously offering dependable device connections. Therefore, using UAVs may eliminate shadowing problems in the wireless communication networks by ensuring LoS in the connections, offering a practical means to serve ground devices effectively [14]. Table 1.1 shows the difference between aerial base stations and terrestrial base stations.

Aerial base stations	Terrestrial base stations
3D deployment	2D deployment
Short term deployment	Long term deployment, enduring
Mobility	Fixed
Good Line-of-Sight	Lack of Line-of-Sight
Unbounded location	Specific location
Limited energy	Unlimited energy

Table 1.1: Aerial base stations and terrestrial base stations.

For instance, by establishing LoS connections with terrestrial nodes, UAVs can offer localization services when ground stations are inaccessible or cannot offer stable connectivity. In line with this strategy, 6G technologies seek to establish ubiquitous connectivity by utilizing UAVs to wirelessly link terrestrial and aerial networks [15].

Along the lines of global coverage plan and fueled through recent advancements in wireless communication technologies, the wide proliferation of IoT devices has anchored IoT in many facets of contemporary life [16]. This rise of Internet of Things technology has resulted in the creation of a wide range of IoT devices. IoT devices are used in smart home apps, tablets, smartphones, edge routers, cellular base stations, smart traffic systems, smart energy meters, linked cars, smart building controls, and more. The fast growth of IoT in various verticals unfolded novel services that require continuous data collection and fast processing of this data for informed decisions. Due to energy and size limitations, IoT devices are often not capable of locally processing complex tasks that might additionally require more data collected from other devices. Edge computing emerged as a viable solution to complement the IoT capabilities rather than cloud computing that results in higher delays and energy consumption. Edge servers that are, however, stationed in fixed positions cannot well adapt to changing environment to ensure a target deadline for given applications. This being said, UAVs are introduced as a mobile platform on which computational resources can be mounted to serve as an agile and flexible edge server. Nonetheless, UAVs can serve as aerial base stations and are projected to significantly improve IoT services by offering more reliable communication links. This is achieved by adjusting the UAVs' locations to establish line of sight connectivity with the available IoT devices, thanks to their high mobility and flexibility [14], [17], [18].

In this study, we aim at combining the benefits of the various computational platforms in a multi-tiered model, where lower tiers are more flexible and closer to IoT devices but possess lower computational resources. Simpler and more delay sensitive tasks tend to be processed at the lower tiers, while more complex tasks and those with relaxed time constraints are offloaded to higher tiers. We consider a field of IoT devices that produce computational tasks with strict time constraints and develop a hierarchical computational model composed of three layers. At the bottom layer, a swarm of UAVs are deployed in a way to establish line-of-sight links and collect the tasks from the IoT devices. The UAVs process part of the tasks and offload the rest to be distributed between middle layer composed of edge servers and the upper layer containing the cloud. The goal is to build a multi-tier computational model composed of a minimized number of UAVs and develop an efficient solution that optimizes the UAV locations, resource allocation, associations, and offloading decisions while ensuring the tasks' deadline.

1.2 Thesis Organization

The rest of this thesis is organized as follows. In Chapter 2, we cover the related literature. Chapter 3 explains the system model and its key components. In Chapter 4, we formulate the considered problem as a mixed integer program. Chapter 5 describes the proposed scalable and efficient solution. Chapter 6 presents performance results for various scenarios. Finally, conclusions are drawn and future works are discussed in Chapter 7.

Chapter 2

Related Literature and Thesis Contributions

Due to their physical size and restricted storage, computational, and energy resources, IoT devices are classified as resource-constrained devices. Allocating their scarce resources while minimizing energy consumption, cost, execution time as well as maximizing computing efficiency become more and more challenging. In particular, these network edge devices are rapidly expanding, which causes large amounts of data to be transmitted continuously in a short amount of time. These data must be rapidly processed and returned.

2.1 UAV-assisted MEC:

Mobile edge computing (MEC) played a major role in assisting IoT devices in performing their computation-intensive tasks within their deadlines at mobile edge servers located at their proximity [19]. In recent years, research on MEC such as power consumption and latency has been actively conducted. The MEC server can handle many tasks, as it gets closer to the device, the processing delays can be significantly reduced compared to cloud servers[20]. Even though MEC offers several benefits, it is still constrained by the positions of fixed towers. As a result, it is difficult to set up MEC anytime or everywhere. Additionally, there is a good chance that natural disasters might occasionally damage the infrastructure. Furthermore, establishing connectivity for temporary usage in rural locations or mountains is very challenging. In the aforementioned conditions, IoT devices are unable to properly function in order to service their users. Therefore, the agility, flexibility and scalability of UAVs has led to the introduction of UAV-assisted MEC to act as a computing platform for IoT devices at dynamic places [21][22]. The UAV-assisted MEC can then speed up computations by shifting tasks to the nearby server which reduces offloading tasks to the cloud, hence, reducing the communication latency and allowing more tasks to be completed within their deadlines.

In [23] and [24], the authors addressed task offloading and trajectory optimization for UAV-enabled MEC networks aiming at minimizing the energy consumption and delay. The authors in [25] jointly optimized the deployment of UAV cloudlets and offloading decisions of tasks with strict latency requirements. In [26], the authors used the UAV as a computing and relay node aiming at reducing the average latency of all users. The authors in [27] proposed a learning-based cooperative particle swarm optimization approach with a Markov random field-based decomposition strategy aiming at optimizing UAV resource allocation while minimizing the maximum response time of forest fire monitoring. In [28], the authors proposed reliability-aware computation offloading in a UAV-enabled MEC system aiming at maximizing the number of requests while optimizing the UAVs deployment and their resource allocation. In [29], a UAV-assisted MEC system in which a set of deployed UAVs attempt to reduce the total task execution time of all the ground users by cooperatively optimizing the proportion of the offloaded tasks, the paths of the UAVs, and the user scheduling variables. Later in [30], the authors studied a UAV-based MEC system in which a moving UAV empowered with a computing server was proposed to assist user equipment in computing their offloaded tasks. By simultaneously optimizing the task-bit allocation and the UAV path, the overall mobile energy consumption was reduced. In [31], authors has analyzed the optimal height of UAV-based base stations for maximum communication coverage. M.Alzenad et al. in [32], proposes an efficient UAV 3D deployment with the goal of maximizing covered users based on optimal altitude and according to various quality of service requirements. In [33], the authors aim at finding the minimum number of UAVs needed and their locations to deliver the different control commands to all the IoT devices while meeting the commands' latency and reliability constraints. Luo in [34] has proposed a cloud-supported UAV application framework that provides UAVs with real-time data processing capabilities by offloading video data to cloud servers. However, offloading data to the remote cloud increases communication latency. The majority of the papers employed various techniques to reduce latency. However, it is equally critical to find a strategy to maximize task execution within the specified time frame.

2.2 Multi-tier computational offloading:

Low latency, significant reliability, and robust online security safety are requirements for many present and emerging applications [35]. These can't be fully satisfied by the standard cloud computing paradigm, which necessitates using high bandwidth links to transfer large amounts of data and computational tasks to the cloud, making it challenging to achieve the demands of low latency and good energy savings. Therefore, to further enhance system performance, research studies are actively working on meeting the massive amount of devices and computation-intensive requests by adopting multi-tier computational offloading. Multi-tier computing was developed in heterogeneous networks to deliver low-latency operations by efficiently merging edge computing with fog computing and taking full advantage of all the resources of the IoT devices, edge servers and cloud [36]

Wang et al. in [37] aimed at minimizing the system latency in heterogeneous multi-layer MEC where tasks are allowed to be offloaded to upper layer more powerful computing servers and cloud. The authors in [38] used machine learning to decide on the cloudlets deployment and resource allocation while minimizing service latency and enhancing resource efficiency. In [39], the authors aimed at maximizing the number of offloading tasks in a three-tier framework composed of multiple users, MECs, and a cloud. The authors in [40] optimized the offloading decision and resources allocations aiming at minimizing the network operator's computational cost and devices' energy consumption usage in a multi-tier edge cloud architecture while respecting the devices' latency requirement. In [41], the authors addressed communication, offloading, and computation resource allocation in a three-layer network architecture that consists of mobile devices, edge and helper cloudlets aiming at minimizing the computation cost and energy consumption. The edge server and cloud combination was used by the majority of the studies. However, it is interesting to investigate the impact of including UAVs into the network as they offer greater LoS and latency.

2.3 Contribution

In contrast to the related work, this thesis presents a UAV-assisted Multi-tier Computing Framework for IoT Networks where the goal is to optimize the number and position of deployed UAVs while maintaining low latency requirements.

The main contributions of this work can be summarized as follows:

1. Developing a multi-tiered computing framework with the lower tier constituted of UAV-mounted cloudlets, and determine the minimum number and optimal placement of UAVs to serve latency-sensitive IoT applications while taking into account a set of limitations related to computational tasks, available resources, and quality of service objectives.

2. In the developed multi-tiered computing framework, we optimize the number and position of deployed UAVs, IoT-to-UAV association, resource allocation,

and task offloading to UAVs, edge servers, and the cloud. To do so, we formulate the problem as a mixed integer programming problem and solving it using successive convex approximation.

3. Propose an efficient solution that decomposes the main problem into two subproblems, the first of which employs three methods: the Elbow technique, the Gap-Statistic method, and the Silhouette method, to find the optimal number of clusters which represents the UAVs. The solution then uses the k-means algorithm to find the positions of UAVs. The output of this subproblems, the number and location of UAVs, will be the input of the second subproblem, whose purpose is to maximize the proportion of admitted tasks. This procedure is repeated by raising the number of UAVs by one until the maximum percentage of admitted tasks is reached.

Chapter 3

System Model

As depicted in Figure 3.1, we consider I IoT devices, denoted by $\mathcal{I} = \{1, 2, ..., I\}$, randomly distributed in an area to routinely monitor and gather data that must be processed for appropriate decision making within a short period of time. The IoT devices are expected to generate λ_i tasks/second, where each task is characterized by a task size s_i, and a deadline T_i. The IoT devices have no computational capabilities to process or partially process the tasks and, thus, need to offload them. Uploading the tasks to the available base stations requires high energy consumption especially in poor channel conditions. To this end, a multi-tier computational model is developed to optimize the task offloading process and deliver the tasks within their target deadline. The computational model is composed of three tiers: UAV-tier, edge-tier, and cloud-tier. The UAV-tier deploys a minimum number of UAVs, denoted by $\mathcal{U} = \{1, 2, ..., U\}$, with strong line-of-sight connection with the IoT devices, and each possess f_u cycles/sec computational capacity. The edge-tier constitutes the available base stations, denoted by $\mathcal{E} = \{1, 2, ..., E\}$, that offer the backhaul links to the UAVs and are in turn equipped with individual edge servers with f_e cycles/sec. The cloud-tier, which is at the upper level, constitutes the cloud resource with a high computational capacity f_c cycles/sec and a fixed upload rate R_c.

In particular, there are three possible situations while computing a task: (1)

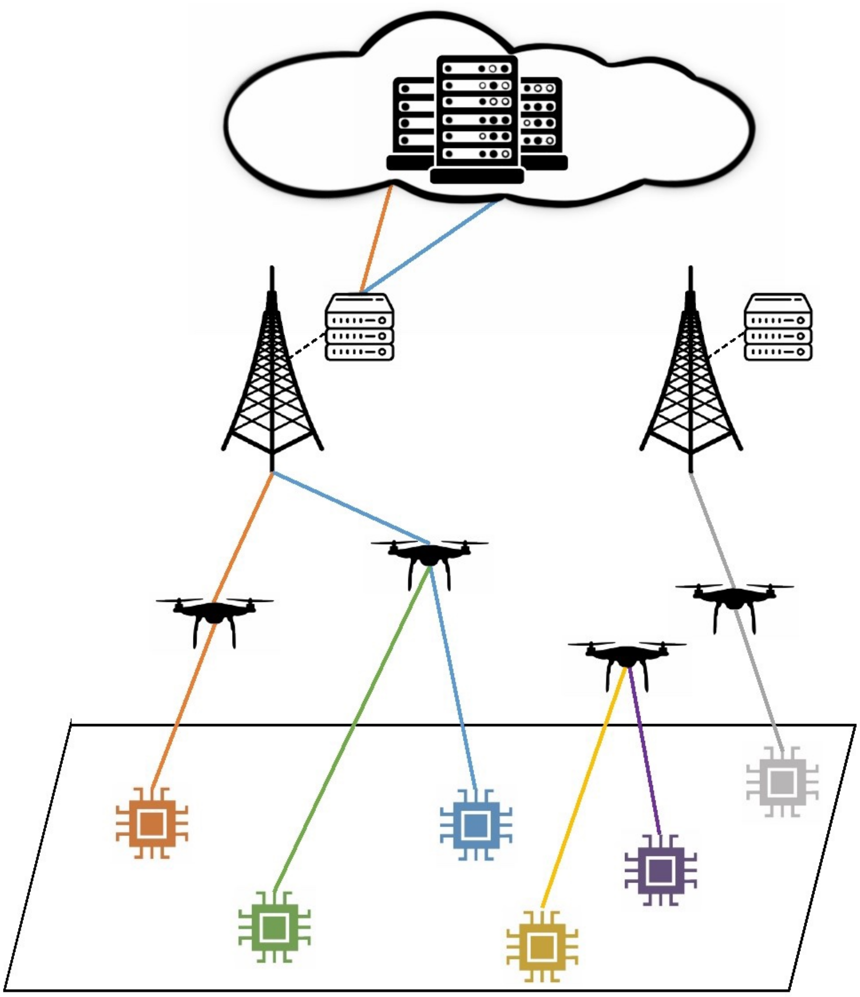


Figure 3.1: System model.

UAV may perform the computations and then send the results to the appropriate IoT device. (2) When the available processing capacity is not enough for the UAV to handle the task, the latter is offloaded to the edge server. The task will be processed by the edge server, and return the result. (3) Finally, when the processing capacity surpasses the capabilities of the edge server, it is routed through the UAV and the edge server to the central cloud.

3.1 Computational Model

Every IoT device $i \in \mathcal{I}$ generates computation requests based on a Poisson distribution with an average rate λ_i. Tasks generated by the same IoT device share a common deadline T_i and offloaded to an associated UAV for potential processing. Each UAV ends up processing a percentage of the workload generated by a number of IoT devices. Since devices $i \in \mathcal{I}$ generate their tasks according to a Poisson process with rate λ_i, the aggregation of workload from different IoT devices can also be modeled as a Poisson process with rate $\sum_i \lambda_i$.

We assume the processing capacity of UAV u in $\mathcal{U}$ is denoted f_u cycles/sec, and the service rate in requests/s is thus computed as f_u/L, where L is the average size of the computation tasks in cycles. The same is assumed for the edge server and the cloud.

UAVs, edge servers and the cloud are assumed to execute IoT requests in an exponentially distributed manner with an average service time equals to $1/\mu_u$, $1/\mu_e$, and $1/\mu_c$ respectively, where μ_u, μ_e and, μ_c are the average service rate of the associated component u, e and, c respectively.

In light of this, every UAV can be modeled as an M/M/1 queuing system with a service rate f_u/L. Similarly, edge servers and the cloud are modeled as M/M/1 queues and collect tasks generated by IoT devices using a Poisson process. therefore, each edge server e in E and the cloud c is characterized as M/M/1 queuing with a service rate f_e/L, and f_c/L, respectively.

3.2 Communication Channel Model

Assuming that the line-of-sight component dominates the wireless channel, the channel gain h between may be stated as follows.

$$h_{xy} = K \left(\frac{\mathrm{d_0}}{\mathrm{d_{x,y}}} \right)^{\alpha} \tag{3.1}$$

where K is the path loss constant (unit-less), α is the path loss exponent which is equal to 2 in free spaces, d_0 is a reference distance (=1m), and $d_{x,y}$ is the distance in meters between the transmitter x and the receiver y. To upload a task from an

IoT device to a UAV, the transmitter x here is the IoT device and the receiver y is the UAV. Same thing is assumed to send task from UAV to edge server.

The attainable data rate of the uplink channel is based on the Free-Space Path Loss model, thus calculated as follows:

-From IoT i to UAV u:

$$R_{iu} = \delta_{iu} B \log_2 \left(1 + \frac{h_{iu} * P_{t,i}}{\sigma^2}\right) \tag{3.2}$$

where δ_{iu} is the number of resource blocks assigned to UAV u from IoT i, B is the bandwidth of one resource block, h_{iu} is the channel model between the IoT device i and UAV u, and $P_{t,i}$ is the transmit power of the IoT device i.

-From UAV u to edge server e:

$$R_{ue} = \delta_{ue} B \log_2 \left(1 + \frac{h_{ue} * P_{t,u}}{\sigma^2}\right) \tag{3.3}$$

where δ_{ue} is the number of resource blocks assigned to edge e from UAV u, B is the bandwidth of one resource block, h_{ue} is the channel model between the UAV u and edge server e, and $P_{t,u}$ is the transmit power of the UAV u.

-For the cloud: We assume a high speed bandwidth of the backhaul link to the cloud. Therefore, similar to [42], the cloud upload rate R_c is a fixed value.

3.3 Delay Model

When the IoT device i offloads its task to one of the UAVs u, then we have a time of t_{iu}^{up} to upload the task. The processing times at UAV u, edge server e, and cloud c are denoted as, t_{iu}^{pro}, t_{ie}^{pro},respectively and t_{ic}^{pro} are the process time required for a task to be processed. In addition, t_{ue}^{up}, and t_{ec}^{up}, are the time required to transmit the task from the UAV to the edge server and from the edge server to the cloud respectively. Additionally, just like in other related works, such as [25] and [43], it is assumed that the task output data will be significantly smaller than the task input data provided by the IoT device. As a result, the download time t_{down} to deliver the response to the IoT device is assumed negligible.

To ensure that the task is executed in one entity only, we define three binary variables m_{iu}, m_{ie}, and m_{ic} where:

$$m_{iu} = \begin{cases} 1, & \text{if task of IoT } i \text{ is executed by UAV } u \\ 0, & \text{otherwise} \end{cases} \tag{3.4}$$

$$m_{ie} = \begin{cases} 1, & \text{if task of IoT } i \text{ is executed by edge } e \\ 0, & \text{otherwise} \end{cases} \tag{3.5}$$

$$m_{ic} = \begin{cases} 1, & \text{if task of IoT } i \text{ is executed by the cloud} \\ 0, & \text{otherwise} \end{cases} \tag{3.6}$$

Where:

$$\sum_{u} m_{iu} + \sum_{e} m_{ie} + m_{ic} = 1; \tag{3.7}$$

Therefore, the total delay t_i^{total} experienced by IoT device i can be calculated as follows:

$$t_i^{total} = t_{iu}^{up} + t_{iu}^{pro} + m_{ie}(t_{ue}^{up}) + t_{ie}^{pro} + m_{ic}(t_{ue}^{up} + t_{ec}^{up}) + t_{ic}^{pro} \tag{3.8}$$

where t_i^{total} should be less than the required deadline T_i of the IoT device i.

We denote by a_{iu}, and b_{ue}, two binary decision variable that indicates if IoT i is associated with UAV u, and if UAV u is associated to edge server e respectively.

$$a_{iu} = \begin{cases} 1, \text{ if IoT } i \text{ is associated to UAV } u \\ 0, \text{ otherwise} \end{cases} \tag{3.9}$$

$$b_{ue} = \begin{cases} 1, \text{ if UAV } u \text{ is associated to edge server } e \\ 0, \text{ otherwise} \end{cases} \tag{3.10}$$

Given that the UAV, edge server, and cloud are each represented as an M/M/1 system as previously mentioned, the process time, which reflects the overall duration spent to process the IoT tasks by the UAV, edge server or the cloud, is determined using Little's law. Therefore, the process time is equal to:

-For UAV:

$$t_{iu}^{pro} = \frac{\mathrm{m_{iu}}}{(\frac{\mathrm{f_u}}{\mathrm{L}} - \sum_{\mathrm{x}} \mathrm{a_{xu}} * \mathrm{m_{xu}} * \lambda_{\mathrm{x}})} \tag{3.11}$$

-For edge server:

$$t_{ie}^{pro} = \frac{\mathrm{m_{ie}}}{(\frac{\mathrm{f_e}}{\mathrm{L}} - \sum_{\mathrm{u}} \sum_{\mathrm{x}} \mathrm{a_{xu}} * \mathrm{b_{ue}} * \mathrm{m_{xe}} * \lambda_{\mathrm{x}})} \tag{3.12}$$

-For cloud:

$$t_{ic}^{pro} = \frac{\mathrm{m_{ic}}}{(\frac{\mathrm{f_c}}{\mathrm{L}} - \sum_{\mathrm{e}} \sum_{\mathrm{u}} \sum_{\mathrm{x}} \mathrm{a_{xu}} * \mathrm{b_{ue}} * \mathrm{m_{xc}} * \lambda_{\mathrm{x}})} \tag{3.13}$$

Chapter 4

Problem Formulation

In this section, we formulate the problem mathematically to find the minimum number of UAVs needed and their locations to serve all IoT devices before the deadline.

The IoT devices transfer their tasks to UAVs. The computations might be handled by the UAVs, which would subsequently transmit the results to the relevant IoT devices. Sometimes, the task is delegated to the edge server when the computational workload cannot be computed at the UAV due to insufficient available computational capacity. The edge server will process the task and deliver the outcome back to the UAV then to the IoT. Finally, the task is routed through the UAV and the edge server to the central cloud when its complexity exceeds the capability of the edge server. The formulated problem establishes the number of UAVs to be deployed, the position of each deployed UAV, the connection of IoT devices to UAVs, UAVs to the edge server, and edge server to the cloud, and the site of task execution. The problem is intended to guarantee that the delay constraint of all devices is met.

We denote by d_u, a binary decision variable that indicates if the UAV u is deployed.

$$d_u = \begin{cases} 1, \text{ if UAV } u \text{ is deployed} \\ 0, \text{ otherwise} \end{cases} \tag{4.1}$$

According to the above, the problem is formulated as a mixed integer program $\mathcal{P}1$ as follows:

$$\text{Minimize} \quad \sum_{u=1}^{U} d_u \tag{4.2}$$

$$\text{subject to} \quad a_{iu} \leq d_u; \ \forall \ i \epsilon [1, I], u \epsilon [1, U] \tag{4.3}$$

$$\sum_{u=1}^{U} a_{iu} = 1; \ \forall \ i \epsilon [1, I], u \epsilon [1, U] \tag{4.4}$$

$$\sum_{e=1}^{E} b_{ue} \leq 1; \ \forall \ u \epsilon [1, U], e \epsilon [1, E] \tag{4.5}$$

$$a_{iu} \geq m_{iu}; \ \forall \ i \epsilon [1, I], u \epsilon [1, U] \tag{4.6}$$

$$\sum_{u} a_{iu} * b_{ue} \geq m_{ie}; \ \forall \ i \epsilon [1, I], e \epsilon [1, E] \tag{4.7}$$

$$\sum_{u} \sum_{e} a_{iu} * b_{ue} \geq m_{ic}; \ \forall \ i \epsilon [1, I] \tag{4.8}$$

$$\begin{gathered} \sum_{i} \sum_{u} \delta_{iu} + \sum_{u} \sum_{e} \delta_{ue} = R \\ \forall \ i \epsilon [1, I], u \epsilon [1, U], e \epsilon [1, E] \end{gathered} \tag{4.9}$$

$$\sum_{u=1}^{U} \sum_{w=1}^{U} \sqrt{((x_u - x_w)^2 + (y_u - y_w)^2} \geq \theta \ \forall \ u, w \epsilon [1, U] \tag{4.10}$$

$$\sum_{u} m_{iu} + \sum_{e} m_{ie} + m_{ic} = 1; \ \forall \ i \epsilon [1, I] \tag{4.11}$$

$$t_i^{total} \leq T_i; \tag{4.12}$$

$$\frac{f_u}{L} - \sum_{i} a_{iu} * m_{iu} * \lambda_i \geq 0; \ \forall \ u \epsilon [1, U] \tag{4.13}$$

$$\frac{f_e}{L} - \sum_u \sum_i a_{iu} * b_{ue} * m_{ie} * \lambda_i \geq 0; \forall\ e\epsilon[1, E] \tag{4.14}$$

$$\frac{f_c}{L} - \sum_e \sum_u \sum_i a_{iu} * b_{ue} * m_{ic} * \lambda_i \geq 0 \tag{4.15}$$

This formulation aims at optimizing the number of deployed UAVs while meeting various constraints related to computational tasks, time limits, available computational resources, and performance constraints.

The objective function in (4.2) minimizes the number of deployed UAVs. This is implemented by just reducing the summation of the whole matrix that indicates whether a UAV u is deployed or not. The constraint in (4.3) ensures that the IoT device i is associated and served to one deployed UAV u. Therefore, if a device is not chosen to be deployed in the network, it is not connected with a UAV. Devices shouldn't be connected to UAV u d_u is set to zero. The binary variable a_{iu} should likewise be set to zero when d_u is zero. In fact, the inequality in equation (4.3) that requires the value of a_{iu} to always be smaller than or equal to d_u makes this possible. On the other hand, a device i can be connected to a UAV u if it is deployed. For instance, a_{iu} is set to one or zero to indicate if a device is connected to UAV u or not, and d_u is set to one.

The constraint in (4.4) enforces each IoT device i to be assigned to only one UAV u. The constraint in (4.5) imposes that each UAV u is assigned to at least one Edge server e. In particular, if the task is executed by the UAV, there is no need for the task to be offloaded to the edge server. For this reason, the association between UAV and edge server might be equal to zero. The constraints (4.6) (4.7) (4.8) ensure that a task of IoT device i cannot be executed by the UAV u or edge server e of the cloud if there is no association between them, respectively. The constraint (4.9) gives the total number of resource blocks that can be assigned for UAVs and edge servers based on the total bandwidth. In particular, R represents the total number of resource block for all the system which is equal to $R = B_t/B$, where B_t is the total bandwidth of the system and B is the total bandwidth of one resource block. δ_{iu} and δ_{ue} are the number of resource blocks allocated for the communication between IoT i and UAV u, and between UAV u and edge server e respectively. Hence, the summation of these resource blocks distributed in the network should be equal to the maximum number of resource blocks allowed. The constraint in (4.10) enforces a minimum safety distance θ between any two deployed UAVs. The constraint in (4.11) imposes that the task is either executed by UAV, edge server, or the cloud. The constraint in (4.12) guarantees that the total delay time t_i^{total} is less than or equal to the required deadline T_i of IoT device i. And finally, the constraints (4.13), (4.14), and (4.15), ensures that the M/M/1 queuing systems at the UAVs, edge servers, and cloud are stable. The formulated problem is a mixed integer non-linear program that is very complex to solve. Also, the constraints (4.10), (4.6), (4.7), (4.8), (4.13), (4.14), and(4.15) are non linear.

The first constraint (4.10) is linearized using Taylor approximation. Suppose we have two UAVs u and v, and x_u, y_u, x_v, y_v correspond to their coordinates respectively. We set x_{u0}, y_{u0}, x_{v0}, and y_{v0} as initial values for the coordinates of UAVs u and v. Therefore, this constraint (4.10) is replaced by

$$\begin{multline} 2 * (x_{u0} - x_{v0})(x_u - x_v) - (x_{u0} - x_{v0})^2 \\ + 2 * (y_{u0} - y_{v0})(y_u - y_v) - (y_{u0} - y_{v0})^2 \geq \theta^2 \end{multline}$$

The other constraints are linearized using typical linearization techniques to eliminate product of two binary variables, or product of binary and continuous variables.

For example, in (4.13), the product of binary variables is replaced by the new binary variable x_{iu}. Then, another continuous variable y_{iu} is created which is equal to the product of the binary variable x_{iu} and the continuous variable λ_i. Suppose we have

$$\mathrm{x}_{iu} = a_{iu}m_{iu}$$

$$\mathrm{y}_{iu} = x_{iu}\lambda_i$$

As a result, the equation (4.13) is replaced by

$$\frac{f_u}{L} - \sum_i y_{iu} \geq 0$$

The same process is applied to (4.14), (4.15), and (4.12). In addition, in (4.12), the t_{up}, and t_{U2E} should be linearized since the rate R_{iu} and R_{ue} in the denominator has the variable δ and the distance as an optimal variables. Therefore, the product of continuous variables x and y is linearized using the McCormicks method where;

if w=xy

The following constraints are added

$$\mathrm{w} \geq x^L y + x y^L - x^L y^L$$

$$\mathrm{w} \geq x^U y + x y^U - x^U y^U$$

$$\mathrm{w} \leq x^U y + x y^L - x^U y^L$$

$$\mathrm{w} \leq x^L y + x y^U - x^L y^U$$

Due to the high number of binary variables, this problem remains complex to solve. In the next section, we propose an efficient decomposition solution to solve the problem

Chapter 5

Proposed Solution: Decomposition Approach

Due to this high complexity of the formulated problem $\mathcal{P}1$, in this section we propose a decomposition approach $\mathcal{P}2$ that defines two subproblems solved iteratively to determine the minimum number of UAVs and their positions, then optimize the IoT-to-UAV association, UAV-to-edge association, resource allocation, and task offloading.

This problem is decomposed into two subproblems, where in the first subproblem $\mathcal{S}1$, we try to minimize the number of UAVs and position them. To determine the minimized number of UAVs capable of covering the complete field of IoT devices, we utilize a clustering technique to find the minimum number of cluster and their positions where each represents a UAV.

Clustering techniques are divided into two main types; partitional clustering and hierarchical clustering. In order to use partitional clustering techniques, such as the k-means algorithm, the number of clusters must be determined beforehand. However, this is not the case with hierarchical clustering. In addition, for big databases, partitional clustering approaches, as opposed to hierarchical clustering, are more effective and have better time complexity. An nxn distance matrix must be computed and stored in order to perform hierarchical clustering. This can be costly and time-consuming for really big datasets. The k-Means technique creates high-quality clusters and is quicker than other clustering algorithms when working with large datasets [44]. For this reason, in our proposed solution, we used some methods and the k-means algorithm to determine the number and position of clusters. Therefore, to determine the number of clusters, first, we employ three methods; the *Elbow method*, the *Gap-Statistic method*, and the *Silhouette method*. Each of these clusters is expected to be served by one UAV. Consequently, after determining the number of UAVs, we deploy k-means to find the optimized coordinates of these UAVs.

The number and position of deployed UAVs are input to the second subproblem $\mathcal{S}2$ that aims at maximizing the percentage of admitted tasks. A task is considered admitted when it is fully processed within the target deadline. If the second subproblem does not yield a situation when all tasks can be processed within their deadline, an additional UAV is deployed and passed to the first subproblem $\mathcal{S}1$ to find the optimized positions that are then fed to the second subproblem $\mathcal{S}2$. This iterates until the workload of all devices are being served or until the problem becomes infeasible. These steps are summarized in the flowchart presented in Figure 5.1.

The Elbow technique (Figure 5.2) varies the amount of clusters k. The WCSS (Within-Cluster Sum of Square) is computed for each value of K. The sum of the squared distances between each point and the centroid of the cluster determines the WCSS value. An elbow is formed by the plot of the calculated WCSS for each k value. As the number of clusters increases, the WCSS will start to decrease. The greatest WCSS value occurs when k=1. When we look at the graph, we can see that it quickly transforms into an elbow at one point. From this point on, the graph starts to travel almost parallel to the X-axis. The best k value, or the most clusters, is the one that corresponds to this point. In particular, we can see that the WCSS decreases when the number of cluster increases. The optimal number of cluster k is the value from where the graph starts to travel almost parallel to the X-axis , which

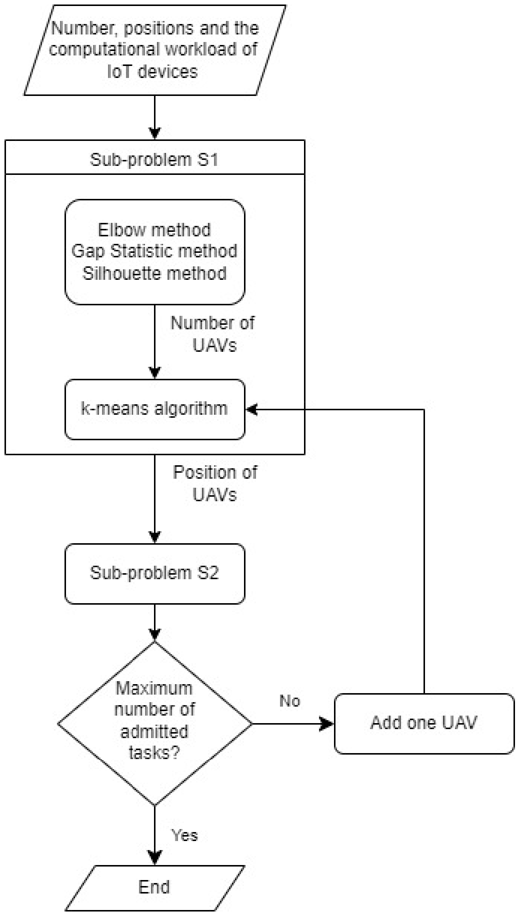


Figure 5.1: Proposed Solution Flowchart

is k=4 in this case.

The gap statistic (Figure 5.3) compares the values that would be anticipated under a null reference distribution for the data with the total intra-cluster variance for various values of k. To determine the ideal clusters, the value that maximizes the gap statistic will be utilized (i.e, that yields the largest gap statistic). In other words, compared to a uniform, randomly dispersed distribution of points, the clustering structure is considerably different. The graph shows that the gap is the highest for k=4 which means that the clustering structure differs significantly from a uniform, randomly distributed point distribution.

The silhouette method (Figure 5.4) evaluates how well a clustering is done. In other words, it establishes how well each object fits within its cluster. A good clustering is indicated by a high average silhouette width. The average silhouette

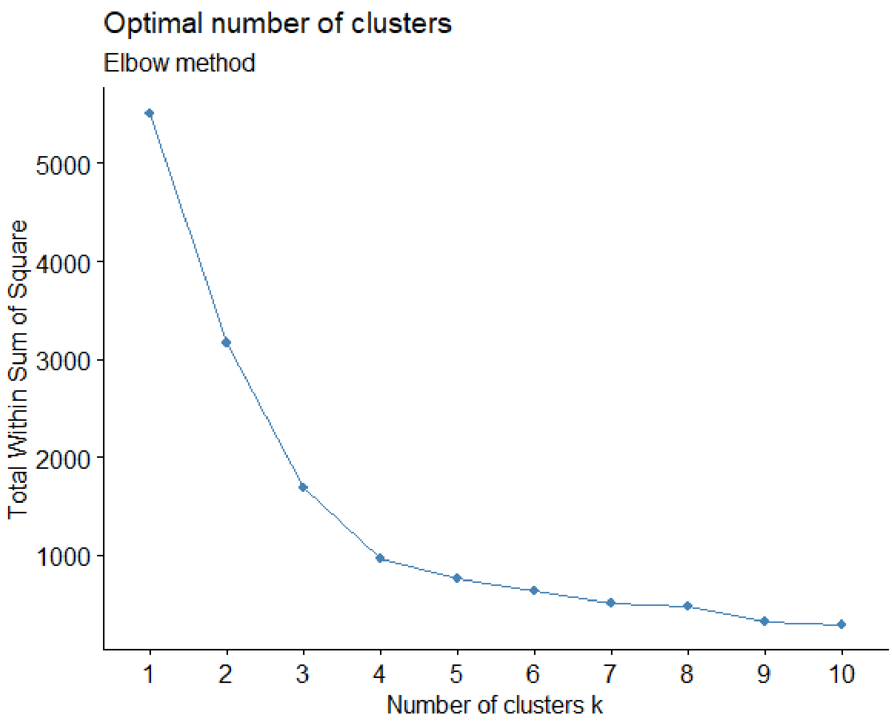


Figure 5.2: Example of the elbow method result.

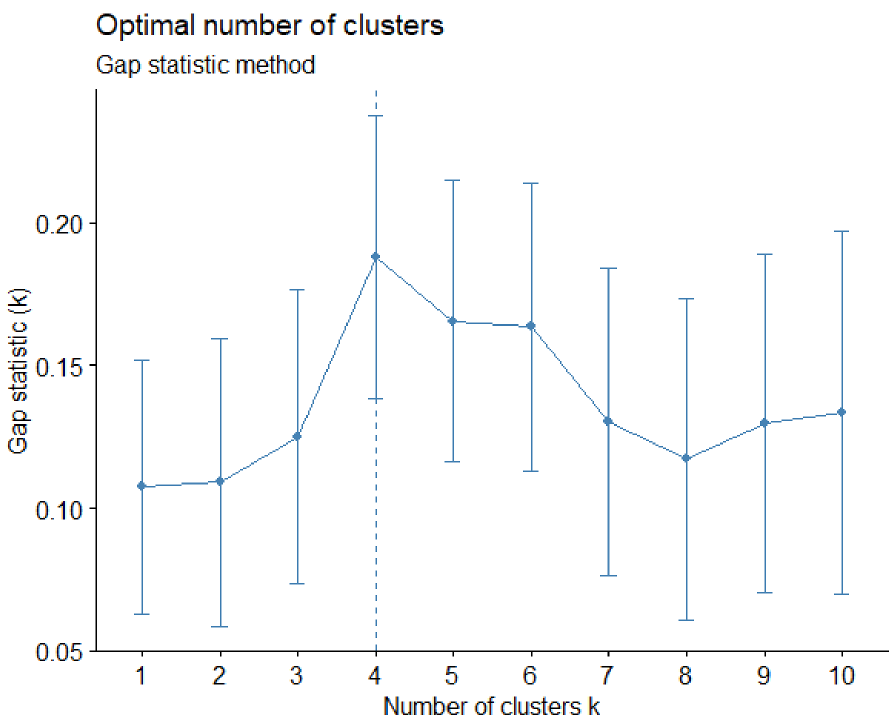


Figure 5.3: Example of the gap statistics method result.

of the observations is calculated using the average silhouette technique for various k values. The number of clusters with the best k throughout a range of probable k values is the one that optimizes the average silhouette. The graph shows the highest average silhouette width for k=2 which means that for this number of cluster each object fits well within its cluster.

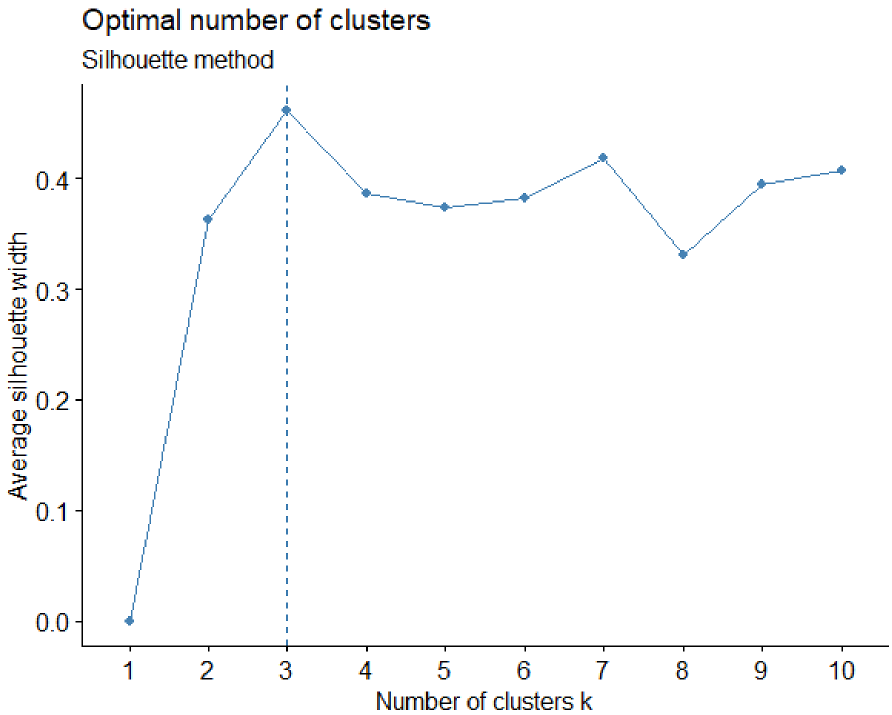


Figure 5.4: Example of the silhouette method result.

Regarding the number of UAVs, each of the applied methods may result in a different value. The average k of the three values is considered and the optimized positions are determined using the k-means algorithm. To guarantee a minimal number of UAVs, we start with $k-1$ UAVs then we run the iterative solution between the two subproblems until the largest workload can be admitted. For example, in the previous plots, Figures 5.2, 5.3, and 5.4 the elbow methods returned k=4, the gap statistics method returned k=4 and the silhouette method returned k=3. Then we take the average approximation of these values. This latter will be used in the k-means algorithm to get the coordinates of the clusters. Figure 5.5 and 5.6 present example scenarios to demonstrate the performance of k-means.

As a consequence, the number and location of the UAVs will be known and used as an input to the second subproblem $\mathcal{P}2$. Since the proposed decomposition solution

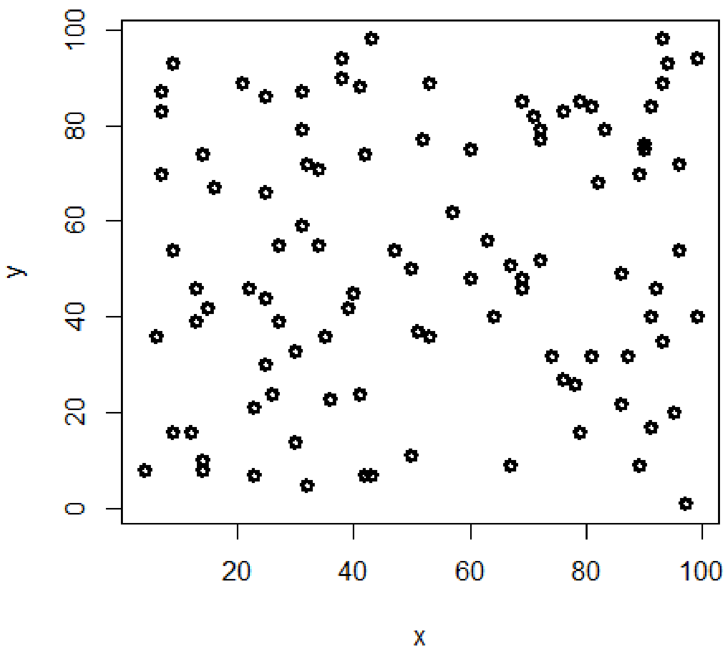


Figure 5.5: Example 100 IoT devices placed in 100 m x 100 m area.

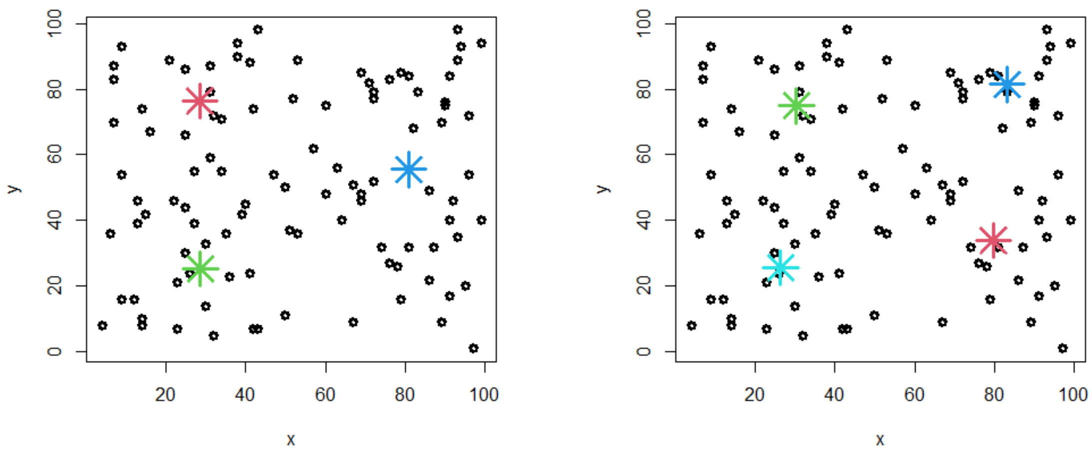


Figure 5.6: Example 100 IoT devices placed in 100 m x 100 m area with the three UAVs (left) and four UAVs (right).

considers two subproblems and a different objective, the problem is reformulated to maximize the summation of the percentage τ_i of the admitted tasks generated by each IoT devices $i \in \mathcal{I}$.

Therefore, the percentage of admitted tasks τ_i is now taken into account while calculating the process time as the following:

-For UAV:

$$t_{iu}^{pro} = \frac{\mathrm{m_{iu}}}{(\frac{\mathrm{f_u}}{\mathrm{L}} - \sum_{\mathrm{x}} \mathrm{a_{xu}} * \mathrm{m_{xu}} * \tau_{\mathrm{x}} * \lambda_{\mathrm{x}})} \tag{5.1}$$

-For edge server:

$$t_{ie}^{pro} = \frac{\mathrm{m_{ie}}}{(\frac{\mathrm{f_e}}{\mathrm{L}} - \sum_{\mathrm{u}} \sum_{\mathrm{x}} \mathrm{a_{xu}} * \mathrm{b_{ue}} * \mathrm{m_{xe}} * \tau_{\mathrm{x}} * \lambda_{\mathrm{x}})} \tag{5.2}$$

-For cloud:

$$t_{ic}^{pro} = \frac{\mathrm{m_{ic}}}{(\frac{\mathrm{f_c}}{\mathrm{L}} - \sum_{\mathrm{e}} \sum_{\mathrm{u}} \sum_{\mathrm{x}} \mathrm{a_{xu}} * \mathrm{b_{ue}} * \mathrm{m_{xc}} * \tau_{\mathrm{x}} * \lambda_{\mathrm{x}})} \tag{5.3}$$

As a result, we can define the new optimization problem as $\mathcal{S}2$:

$$\text{Maximize} \quad \sum_{i=1}^{I} \tau_i \lambda_i \tag{5.4}$$

$$\text{subject to} \quad \sum_{u=1}^{U} a_{iu} = 1; \ \forall \ i \epsilon [1, I], u \epsilon [1, U] \tag{5.5}$$

$$\sum_{e=1}^{E} b_{ue} \leq 1; \ \forall \ u \epsilon [1, U], e \epsilon [1, E] \tag{5.6}$$

$$a_{iu} \geq m_{iu}; \ \forall \ i \epsilon [1, I], u \epsilon [1, U] \tag{5.7}$$

$$\sum_{u} a_{iu} * b_{ue} \geq m_{ie}; \ \forall \ i \epsilon [1, I], e \epsilon [1, E] \tag{5.8}$$

$$\sum_{u} \sum_{e} a_{iu} * b_{ue} \geq m_{ic}; \ \forall \ i \epsilon [1, I] \tag{5.9}$$

$$\sum_i \sum_u \delta_{iu} + \sum_u \sum_e \delta_{ue} = R; \forall\ i\epsilon[1, I], u\epsilon[1, U], e\epsilon[1, E] \tag{5.10}$$

$$\sum_u m_{iu} + \sum_e m_{ie} + m_{ic} = 1;\ \forall\ i\epsilon[1, I] \tag{5.11}$$

$$t_i^{total} \leq T_i;\ \forall\ i\epsilon[1, I], u\epsilon[1, U], e\epsilon[1, E] \tag{5.12}$$

$$\frac{f_u}{L} - \sum_i a_{iu} * m_{iu} * \tau_i * \lambda_i \geq 0\ \forall\ u\epsilon[1, U] \tag{5.13}$$

$$\frac{f_e}{L} - \sum_u \sum_i a_{iu} * b_{ue} * m_{ie} * \tau_i * \lambda_i \geq 0 \forall\ e\epsilon[1, E] \tag{5.14}$$

$$\frac{f_c}{L} - \sum_e \sum_u \sum_i a_{iu} * b_{ue} * m_{ic} * \tau_i * \lambda_i \geq 0 \tag{5.15}$$

This formulation aims at maximizing the number of admitted tasks while meeting various constraints related to computational requirements, time limits, available computational resources, and performance constraints. The goal of this formulation (5.4) in $\mathcal{S}2$ is to maximize the percentage of tasks that are fully processed within their target deadline. This is implemented by just maximizing the product of the percentage of admitted tasks τ_i and the task generation rates λ_i of all IoT devices $i \in \mathcal{I}$.

The formulated problem is mixed integer non-linear problem due to the constraints (5.7), (5.8), (5.9), (5.13), (5.14), and(5.15). These constraints have been linearized as described in previous section.

Chapter 6

Simulation Results and Performance Analysis

In this section, we examine the performance of problem $\mathcal{P}1$ solution and the decomposition solution $\mathcal{P}2$ in terms of efficiency and scalability. To do so, we present detailed simulation results and evaluate the performance in terms of the number of deployed UAVs and time complexity. Then, we apply the proposed decomposition solution to selected application use cases by considering different time limits for the computational tasks. The decomposition problem is addressed using the mixed integer linear programming solver of MATLAB's optimization toolbox after being linearized as mentioned earlier, the three methods and the k-means algorithm are implemented using the R programming language.

In our simulation, we consider a small network with a 100 m x 100 m area and randomly dispersed IoT devices. The reason behind choosing a small network is that the time complexity is very high for the solution of problem $\mathcal{P}1$. The system parameters are summarized in Table 6.1 unless otherwise indicated.

We begin by evaluating the solutions to $\mathcal{P}1$ and $\mathcal{P}2$. The average number of deployed UAVs in relation to the total number of IoT devices is presented in Figure 6.1. We can see that the decomposition solution produces results that are very

Parameters	Value
IoT transmit power $P_{t,i}$	0.2 Watt
UAV transmit power $P_{t,u}$	5 Watts
UAV computational capacity f_u	1.5 GHz
Edge server computational capacity f_e	10 GHz
Cloud computational capacity f_c	1 THz
Total bandwidth B_t	20 MHz
Path loss constant K	1
Path loss exponent α	2
Reference distance d_0	1 m
Thermal noise power σ^2	10^-6 Watts
Task size s_i	50 KB
Task computational demand L	250 cycles/bit
Task deadline T_i	10 ms

Table 6.1: System parameters used in the simulations.

close to the solution of $\mathcal{P}1$. We may also deduce with certainty that more UAVs are required to complete all tasks when more devices are added, leading to a rise in the number of requests. Additionally, we note that we stopped the simulations at 100 devices due to extensively increased execution time to solve $\mathcal{P}1$.

Figure 6.2, an increase in IoT devices creates a bigger network situation and dramatically affects how quickly the solution of $\mathcal{P}1$ may be generated. On the other hand, the proposed approach is demonstrated to be a very effective alternative to produce highly optimized results with low complexity for reasonably large networks. For example, the decomposition solution demonstrated to be 22 times faster than the solution of $\mathcal{P}1$, while only requiring one more UAVs for a network with 100 IoT devices.

Due to the problem $\mathcal{P}1$ solution's high level of complexity, we perform our next simulations utilizing only the decomposition solution. Therefore, we increased the network size to 500 m x 500 m for all the next simulations, and the number of IoT devices is set to 100 devices.

To further study the effects of time deadlines and response times we plot in Figure 6.3 the average number of deployed UAVs for different verticals and according

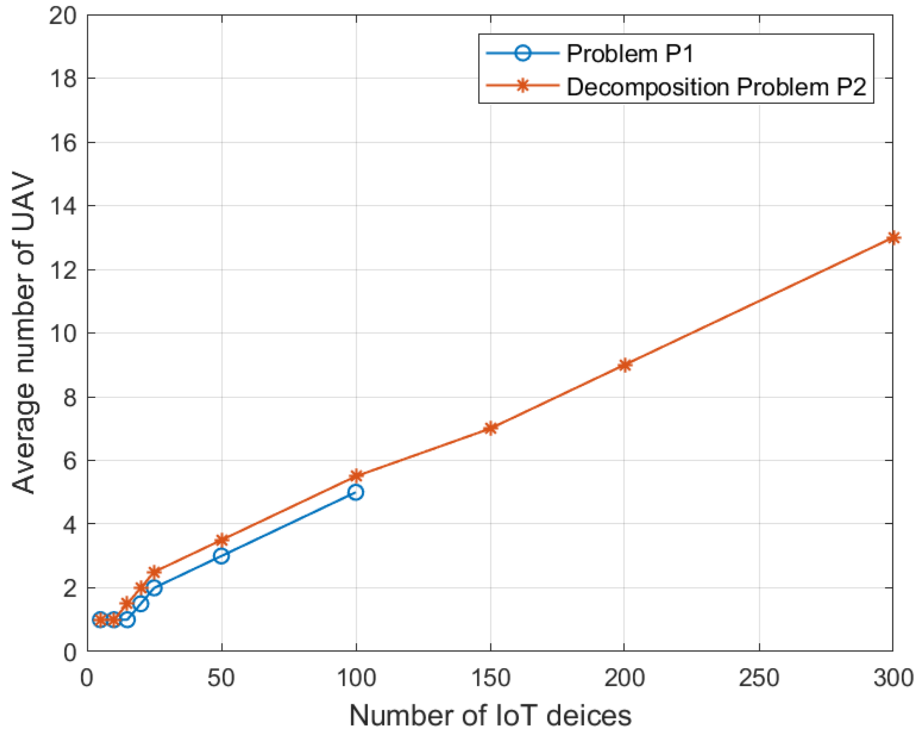


Figure 6.1: Average number of UAVs required to serve the IoT devices for problem $\mathcal{P}1$ solution compared to the decomposition problem $\mathcal{P}2$ solution, as function of total number of devices.

to different request rates. The computation task request rate ranges between 10 and 100 requests/sec. The graph demonstrates how the number of deployed UAVs rises along with the request rate. Due to the low computational capacity of the UAVs, we can observe that the number of UAVs decreases when the request rate increases. Additionally, as the deadline increases, the number of UAVs deployed becomes constant because they are mainly employed as a relay between the devices and the edge server and most tasks will be executed in the cloud.

Industry Vertical	Deadline Range	Applied Latency Limit
Factory Automation	0.25 - 10 ms	2 ms
Manufacturing Cell	5 ms	5 ms
Smart Grid	3 - 20 ms	10 ms
Road Safety	10 - 100 ms	20 ms

Table 6.2: IoT industry verticals with their deadline ranges as per [45].

For the following simulations, we consider different applications set up on the

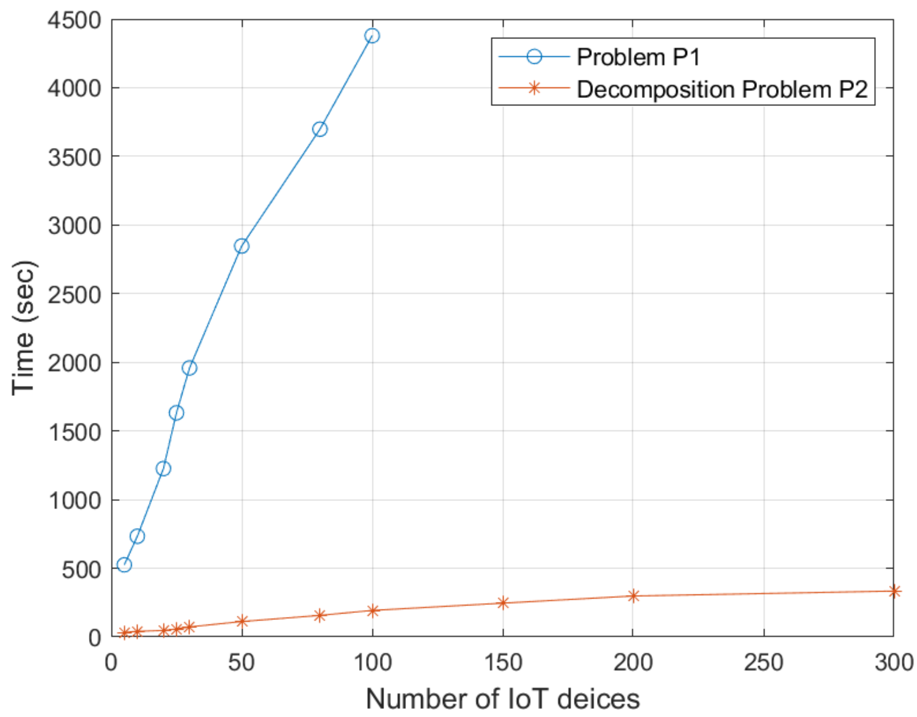


Figure 6.2: Execution time of problem $\mathcal{P}1$ solution versus the decomposition problem $\mathcal{P}2$ solution as a function of the number of IoT devices.

IoT devices. We take into account diverse industrial verticals with varying time constraints and distinctive features. Table 6.2 lists the industry verticals that were used in this section. We contrast the four various IoT industrial sectors while accounting for their latency constraints.

In Figure 6.4, we plot the average number of UAVs required to serve the IoT devices as a function of the deadline for different task input sizes. The graph shows that, when the task sizes increases, the average number of UAVs also increases. This is due to the high upload time with a small deadline. For a given deadline, when the total time increases, the system will need more UAVs to execute the tasks before deadline. In addition, since executing tasks by the edge server and the cloud takes more time than executing them by the UAVs, when the deadline increases, we can see that the number of UAVs decreases.

In Figure 6.5, we study the offloading distribution across different industries for various request rates. The graph shows that when the deadline is low, none of the tasks is offloaded to the cloud, most tasks are executed by the UAVs and the edge

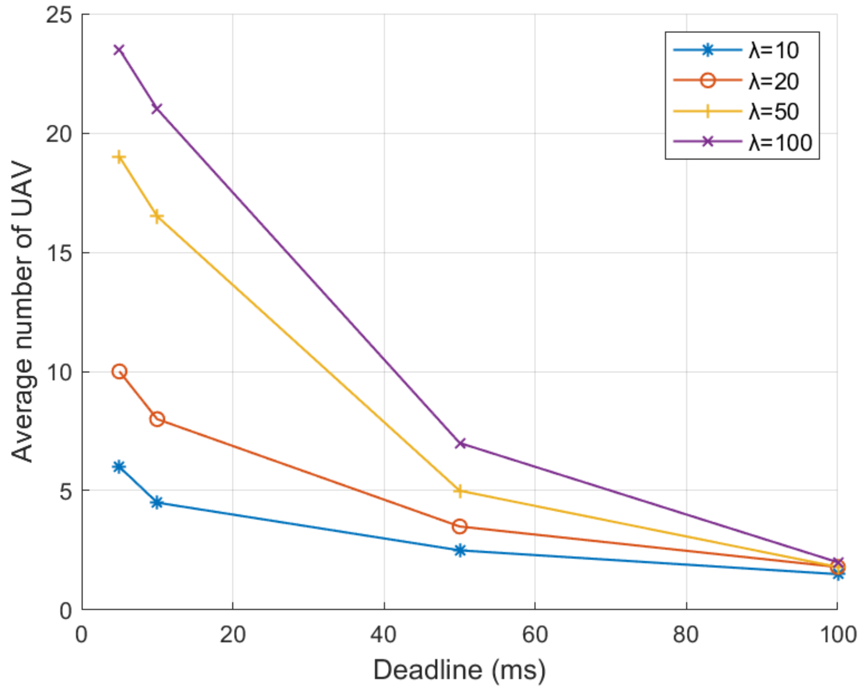


Figure 6.3: Average number of UAVs required to serve the IoT devices based on the decomposition problem solution as a function of the deadline for different rate of computation task requests.

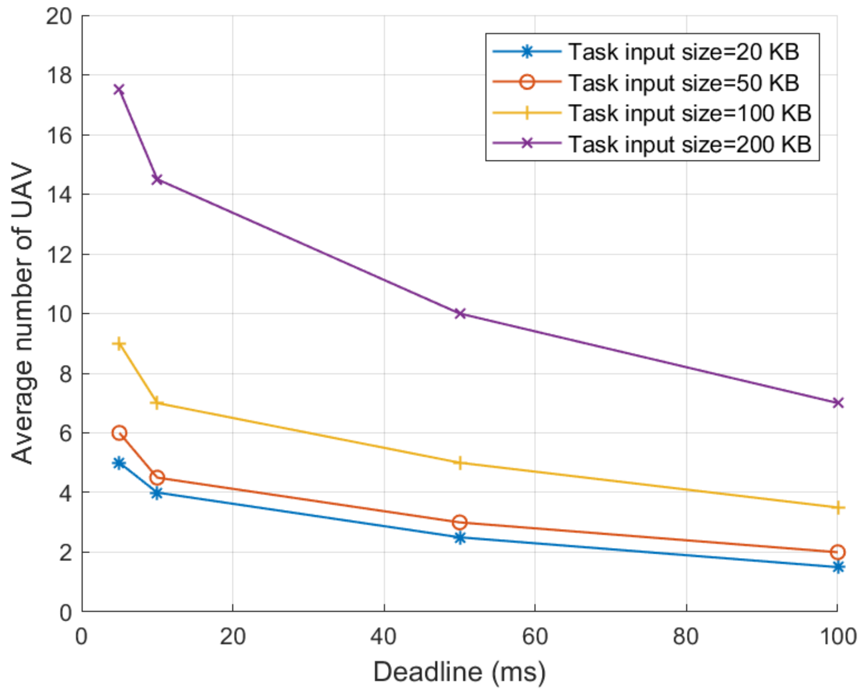


Figure 6.4: Average number of UAVs required to serve the IoT devices based on the decomposition problem solution as a function of the deadline for different task input sizes.

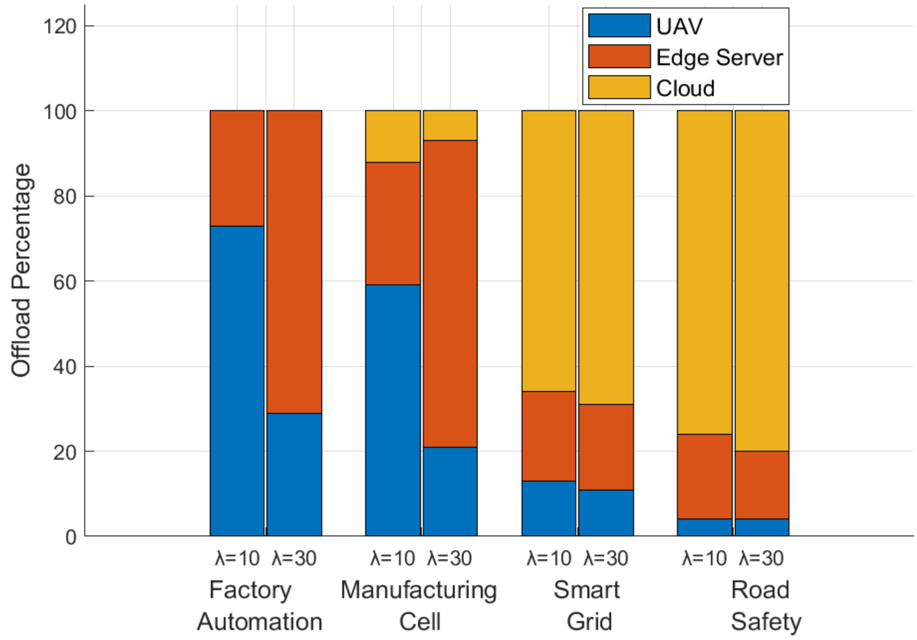


Figure 6.5: Offload percentage to serve IoT devices in different industry verticals as a function of the rate of computation task requests.

server. In addition, it is showed that as the the request rate increases, more tasks are offloaded to the edge server and the cloud. This is due to the low computation capacity of the UAVs. Additionally, when the deadline increases, we can observe that more tasks are executed by the cloud.

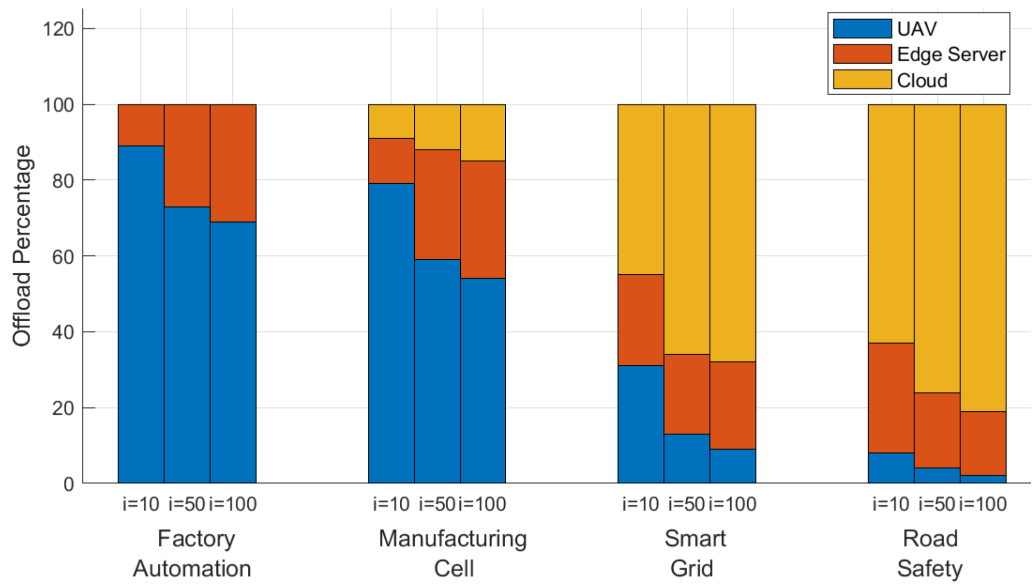


Figure 6.6: Offload percentage to serve IoT devices in different industry verticals as a function of different number of devices.

To further study the offload percentage to serve IoT devices, we plot Figure 6.6 to examine the offload distribution across various industries and for different number of IoT devices. In this case, the computation task request rate is fixed for all devices and set to 10 requests/sec. As the previous plot showed, a tight deadline leads to executing most tasks by the UAVs and increasing the deadline results in offloading the tasks to the edge server or the cloud. Moreover, the graph illustrated that increasing the number of IoT devices leads to offloading more tasks to the edge server or the cloud because the UAVs cannot support the high amount of tasks most of them will be employed as a relay. Similarly, when the deadline is tight and the number of devices is low, most tasks are offloaded to the UAVs due to its computation capacity.

Chapter 7

Conclusion and Future Work

In this thesis, we studied the problem a multi-tier computational model where the goal is to optimize the number and position of deployed UAVs while maintaining a low latency requirement. The problem was formulated as a mixed integer program, and we suggested a low complexity effective solution by splitting it into two sub-problems. In the first sub-problem, we used three different methods to minimize the number of UAVs and the k-means algorithm to determine their positions, and in the second one, we use these results as an input and keep on iterating by increasing the number of UAVs until we reach the maximum number of admitted tasks. Results are shown and examined in relation to a variety of criteria and quality of service restrictions. We showed that the suggested method works well in terms of reaching near-optimal performance with very little complexity, making it particularly interesting for scenarios involving significant number of IoT devices.

For future work, we can consider the IoT devices' mobility as though they were mobiles. Thus, the UAVs' mobility must also be taken into account. In addition, UAVs use a variety of power sources, but each has different limitations in terms of weight distribution, charging and discharging duration, size, payload capacity, energy density, and power density. Therefore, it is also interesting to study the energy of the UAVs in a network.

Bibliography

[1] M. Giordani, M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi, "Toward 6g networks: Use cases and technologies," *IEEE Communications Magazine*, vol. 58, no. 3, pp. 55–61, 2020.

[2] P.-Y. Qin, L.-Z. Song, and Y. J. Guo, "Conformal transmitarrays for unmanned aerial vehicles aided 6g networks," *IEEE Communications Magazine*, vol. 60, no. 1, pp. 14–20, 2022.

[3] Y. Spyridis, T. Lagkas, P. Sarigiannidis, *et al.*, "Towards 6g iot: Tracing mobile sensor nodes with deep learning clustering in uav networks," *Sensors*, vol. 21, no. 11, p. 3936, 2021.

[4] K. P. Valavanis and G. J. Vachtsevanos, *Handbook of unmanned aerial vehicles.* Springer, 2015, vol. 1.

[5] C. A. Wargo, G. C. Church, J. Glaneueski, and M. Strout, "Unmanned aircraft systems (uas) research and future analysis," in *2014 IEEE Aerospace Conference*, IEEE, 2014, pp. 1–16.

[6] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A tutorial on uavs for wireless networks: Applications, challenges, and open problems," *IEEE communications surveys & tutorials*, vol. 21, no. 3, pp. 2334–2360, 2019.

[7] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Communications magazine*, vol. 54, no. 5, pp. 36–42, 2016.

[8] B. Galkin, J. Kibilda, and L. A. DaSilva, "Uavs as mobile infrastructure: Addressing battery lifetime," *IEEE Communications Magazine*, vol. 57, no. 6, pp. 132–137, 2019.

[9] S. Morton, R. D'Sa, and N. Papanikolopoulos, "Solar powered uav: Design and experiments," in *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, IEEE, 2015, pp. 2460–2466.

[10] A. S. Abdalla and V. Marojevic, "Machine learning-assisted uav operations with the utm: Requirements, challenges, and solutions," in *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, IEEE, 2020, pp. 1–5.

[11] F. Achermann, N. R. Lawrance, R. Ranftl, A. Dosovitskiy, J. J. Chung, and R. Siegwart, "Learning to predict the wind for safe aerial vehicle planning," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 2311–2317.

[12] T. Larrabee, H. Chao, M. Rhudy, Y. Gu, and M. R. Napolitano, "Wind field estimation in uav formation flight," in *2014 American Control Conference*, IEEE, 2014, pp. 5408–5413.

[13] Z. Xiao and Y. Zeng, "An overview on integrated localization and communication towards 6g," *Science China Information Sciences*, vol. 65, no. 3, pp. 1–46, 2022.

[14] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile unmanned aerial vehicles (uavs) for energy-efficient internet of things communications," *IEEE Transactions on Wireless Communications*, vol. 16, no. 11, pp. 7574–7589, 2017.

[15] M. Latva-aho, K. Leppänen, F. Clazzer, and A. Munari, "Key drivers and research challenges for 6g ubiquitous wireless intelligence," 2020.

[16] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, "Federated learning: A survey on enabling technologies, protocols, and applications," *IEEE Access*, vol. 8, pp. 140 699–140 725, 2020.

[17] S. Sharafeddine and R. Islambouli, "On-demand deployment of multiple aerial base stations for traffic offloading and network recovery," *Computer Networks*, vol. 156, pp. 52–61, 2019.

[18] M. Samir, D. Ebrahimi, C. Assi, S. Sharafeddine, and A. Ghrayeb, "Leveraging uavs for coverage in cell-free vehicular networks: A deep reinforcement learning approach," *IEEE Transactions on Mobile Computing*, vol. 20, no. 9, pp. 2835–2847, 2020.

[19] L. A. Haibeh, M. C. E. Yagoub and A. Jarray, "A Survey on Mobile Edge Computing Infrastructure: Design, Resource Management, and Optimization Approaches," *IEEE Access*, vol. 10, pp. 27 591–27 610, 2022.

[20] J. Ren, D. Zhang, S. He, Y. Zhang, and T. Li, "A survey on end-edge-cloud orchestrated network computing paradigms: Transparent computing, mobile edge computing, fog computing, and cloudlet," *ACM Computing Surveys (CSUR)*, vol. 52, no. 6, pp. 1–36, 2019.

[21] P. McEnroe, S. Wang and M. Liyanage, "A Survey on the Convergence of Edge Computing and AI for UAVs: Opportunities and Challenges," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 15 435–15 459, 2022.

[22] L. Zhang, Z.-Y. Zhang, L. Min, *et al.*, "Task offloading and trajectory control for uav-assisted mobile edge computing using deep reinforcement learning," *IEEE Access*, vol. 9, pp. 53 708–53 719, 2021.

[23] B. Liu, Y. Wan, F. Zhou, Q. Wu, and R. Q. Hu, "Resource allocation and trajectory design for miso uav-assisted mec networks," *IEEE Transactions on Vehicular Technology*, 2022.

[24] H. Z. D. Wang J. Tian and D. Wu, "Task Offloading and Trajectory Scheduling for UAV-Enabled MEC Networks: An Optimal Transport Theory Perspective," *IEEE Wireless Communications Letters*, vol. 11, no. 1, pp. 150–154, 2022.

[25] R. Islambouli and S. Sharafeddine, "Optimized 3D Deployment of UAV-Mounted Cloudlets to Support Latency-Sensitive Services in IoT Networks," *IEEE Access*, vol. 7, pp. 172 860–172 870, 2019.

[26] L. Zhang and N. Ansari, "Latency-Aware IoT Service Provisioning in UAV-Aided Mobile-Edge Computing Networks," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10 573–10 580, 2020.

[27] L. Sun, L. Wan and X. Wang, "Learning-Based Resource Allocation Strategy for Industrial IoT in UAV-Enabled MEC Systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 5031–5040, 2021.

[28] E. E. Haber, H. A. Alameddine, C. Assi and S. Sharafeddine, "A Reliability-Aware Computation Offloading Solution via UAV-mounted Cloudlets," 2019.

[29] Q. Hu, Y. Cai, G. Yu, Z. Qin, M. Zhao, and G. Y. Li, "Joint offloading and trajectory design for uav-enabled mobile edge computing systems," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1879–1892, 2018.

[30] S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a uav-mounted cloudlet: Optimization of bit allocation and path planning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2049–2063, 2017.

[31] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal lap altitude for maximum coverage," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569–572, 2014.

[32] M. Alzenad, A. El-Keyi, F. Lagum, and H. Yanikomeroglu, "3-d placement of an unmanned aerial vehicle base station (uav-bs) for energy-efficient maximal

coverage," *IEEE Wireless Communications Letters*, vol. 6, no. 4, pp. 434–437, 2017.

[33] N. Abbas, A. Mrad, A. Ghazleh, and S. Sharafeddine, "Uav-based relay system for iot networks with strict reliability and latency requirements," *IEEE Networking Letters*, 2021.

[34] C. Luo, J. Nightingale, E. Asemota, and C. Grecos, "A uav-cloud system for disaster sensing applications," in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, IEEE, 2015, pp. 1–5.

[35] S. Andreev, V. Petrov, K. Huang, M. A. Lema and M. Dohler, "Dense Moving Fog for Intelligent IoT: Key Challenges and Opportunities," *IEEE Communications Magazine*, vol. 57, no. 5, pp. 34–41, 2019.

[36] T. Koketsu Rodrigues, J. Liu and N. Kato, "Offloading Decision for Mobile Multi-Access Edge Computing in a Multi-Tiered 6G Network," *IEEE Transactions on Emerging Topics in Computing*, pp. 1–1, 2021.

[37] B. D. P. Wang Z. Zheng and L. Song, "HetMEC: Latency-Optimal Task Assignment and Resource Allocation for Heterogeneous Multi-Layer Mobile Edge Computing," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4942–4956, 2019.

[38] T. K. Rodrigues, K. Suto and N. Kato, "Edge Cloud Server Deployment With Transmission Power Control Through Machine Learning for 6G Internet of Things," *IEEE Transactions on Emerging Topics in Computing*, 2019.

[39] S. Tian, C. Chi, S. Long, S. Oh, Z. Li and J. Long, "User Preference-Based Hierarchical Offloading for Collaborative Cloud-Edge Computing," *IEEE Transactions on Services Computing*, 2021.

[40] E. El Haber, T. M. Nguyen and C. Assi, "Joint Optimization of Computational Cost and Devices Energy for Task Offloading in Multi-Tier Edge-Clouds," *IEEE Transactions on Communications*, vol. 67, no. 5, pp. 3407–3421, 2019.

[41] N. Li, S. Yang, Z. Wang, W.Hao and Y. Zhu, “Multi-Tier MEC Offloading Strategy Based on Dynamic Channel Characteristics,” *IET Communications*, vol. 14, no. 22, pp. 4029–4037, 2020.

[42] M.-H. Chen, B. Liang, and M. Dong, “Joint offloading decision and resource allocation for multi-user multi-task mobile cloud,” in *2016 IEEE International Conference on Communications (ICC)*, IEEE, 2016, pp. 1–6.

[43] X. Sun and N. Ansari, “Latency aware workload offloading in the cloudlet network,” *IEEE Communications Letters*, vol. 21, no. 7, pp. 1481–1484, 2017.

[44] M. Kaushik and B. Mathur, “Comparative study of k-means and hierarchical clustering techniques,” *International Journal of Software & Hardware Research in Engineering*, vol. 2, no. 6, pp. 93–98, 2014.

[45] P. Schulz, M. Matthe, H. Klessig, *et al.*, “Latency critical iot applications in 5g: Perspective on the design of radio interface and network architecture,” *IEEE Communications Magazine*, vol. 55, no. 2, pp. 70–78, 2017.