

Received November 12, 2019, accepted November 22, 2019, date of publication November 27, 2019, date of current version December 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2956150

# Optimized 3D Deployment of UAV-Mounted Cloudlets to Support Latency-Sensitive Services in IoT Networks

**LANIA ISLAMBOULI<sup>1</sup>** AND **SANAA SHARAFEDDINE<sup>1</sup>**, (Senior Member, IEEE)

Department of Computer Science and Mathematics, Lebanese American University, Beirut 1102 2801, Lebanon

Corresponding author: Sanaa Sharafeddine (sanaa.sharafeddine@lau.edu.lb)

This work was supported by a grant from the Research and Development Council of the School of Arts and Sciences, Lebanese American University.

**ABSTRACT** The paradigm of Internet of Things (IoT) is transforming physical environments into smart and interactive platforms to offer a wide range of innovative services supported by the evolution towards 5G networks. A major class of emerging services relies on highly intensive computations to make real-time decisions with ultra-low latency. Edge computing has been established as an effective approach to reduce the latency overhead of cloud computing and effectively augment the computational capabilities of IoT devices. In this work, we leverage the mobility and agility of Unmanned Aerial Vehicles (UAVs) as mobile edge servers or cloudlets to offer computation offloading opportunities to IoT devices. In particular, we consider the joint problem of optimizing the number and positions of deployed UAV cloudlets in 3D space and task offloading decisions with cooperation among UAVs, in order to provision IoT services with strict latency requirements. We formulate the problem as a mixed integer program, and propose an efficient meta-heuristic solution based on the ions motion optimization algorithm. The performance of the meta-heuristic solution is evaluated and compared to the optimal solution as a function of various system parameters and for different application use cases. It is shown to achieve near-optimal performance with low complexity and, thus, can efficiently scale up to large IoT network scenarios.

**INDEX TERMS** Edge computing, unmanned aerial vehicles, 3D deployment, IoT network.

## I. INTRODUCTION

The development of 5G networks has fostered the advancement of numerous applications including augmented reality, autonomous driving, and remote operation systems. IoT devices are becoming an integral part of various industries to facilitate latency-sensitive automated services by continuously monitoring environments, performing advanced computations, and making critical decisions. IoT devices can fall short of the needed capabilities and, thus, do regularly rely on offloading computations to remote servers [1], [2]. Multi-access edge computing (MEC) has emerged as a promising alternative solution to cloud computing to process latency-sensitive computational requests closer to resource-constrained IoT devices in order to avoid long distance transmissions that consume time and energy [3]. In addition to the direct benefits experienced by the devices,

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Maaz Rehan<sup>1</sup>.

MEC saves the backhaul infrastructure and the core network from significant traffic load especially in dense network scenarios [4].

IoT networks are typically prone to dynamic changes and continuous shifts in device density, positions, and types of requested services. Hence, MEC solutions in such environments have to adapt their resources and capabilities to meet the changing demands. In this work, we propose a dynamic MEC system to serve a given IoT network running latency-sensitive services. The proposed MEC system is provisioned by cloudlets (or edge servers) mounted on Unmanned Aerial Vehicles (UAVs) that hover close to the IoT devices to process their generated workload within target latency requirements. As opposed to wireless fixed cloudlets, UAV-mounted cloudlets yield an efficient and relatively low cost solution that can be flexibly deployed in dynamic environments, especially in ultra-dense networks or remote hard-to-reach areas. Owing to their high agility and mobility, UAVs present themselves as an effective and prompt means

in enabling various communication and computing services for a wide range of industry verticals [5], [6]. They allow edge servers to fly closer to IoT devices establishing better line of sight (LoS) links leading to higher transmission rate and consequently reduced energy consumption. Moreover, additional UAVs can be deployed if needed in response to a high surge in traffic demand.

The problem of deploying UAV-mounted cloudlets to support latency-sensitive services in IoT networks is inherently composed of multiple subproblems: 1) planning the network of UAV-mounted cloudlets to determine the optimized number and 3D positions of the UAVs needed to serve the given demand; 2) associating IoT devices to the deployed UAVs, where each UAV may either provide edge computing capability or serve as a relay node to other UAVs that provide edge computing capability; 3) distributing the generated IoT workload among the deployed UAV-mounted cloudlets. In this work, we formulate this joint problem as a mixed integer program (MIP) with the key objective of minimizing the number of deployed UAVs while serving all IoT workload within its required deadline. After demonstrating that the problem is NP-hard, we propose an efficient solution based on the meta-heuristic ions motion optimization algorithm [7]. We demonstrate the effectiveness of the proposed solution using extensive simulations with comparisons to the optimal solution as a function of various system parameters.

The rest of this paper is organized as follows. In Section II, we cover the related literature and highlight the main contributions of this work. Section III details the system model and its key components. In Section IV, we formulate the considered problem as a mixed integer program. Section V describes the proposed scalable and efficient solution based on the ions motion algorithm. Section VI presents performance results for various scenarios. Finally, conclusions are drawn in Section VII.

## II. RELATED LITERATURE

### A. MOBILE EDGE COMPUTING

Mobile edge computing has been shown to be an effective solution to deal with the downsides of the power and computational constraints in IoT devices [8], [9]. To this end, a wide range of research questions need to be addressed for each possible network scenario depending on the performance objectives and implementation constraints.

For example, in [10] the authors studied the problem of latency-aware workload offloading in edge computing. They suggested an architecture based on software defined networks, and formulated the problem with the goal of minimizing the response time of the end users. They proved that this problem is NP-hard by a reduction from the known partition problem. They suggested a heuristic algorithm to solve the formulated problem based on a greedy strategy. Another work in [11] addressed the joint optimization problem of resource allocation and user requests' offloading in mobile

edge computing networks. In addition, they formulated the problem to minimize the utilized power by the IoT devices and proposed an efficient heuristic algorithm to solve the optimization problem based on an iterative approach. The authors in [12] studied the planning of an edge cloud network in mobile access networks. They focused on the placement of access points and cloudlets, together with the assignment of users to cloudlets. They considered two design cases, the first being a static state network and the second being a dynamic state network with varying service and load level due to user mobility. Authors performed extensive simulations using real 4G cellular datasets to evaluate the tradeoffs resulting from the two design cases. In [13], an end-to-end hybrid cloudlet placement architecture over passive optical networks was proposed. The authors formulated the problem of cost-optimized cloudlet placement as a mixed-integer non-linear programming problem while achieving delay requirements suitable for tactile services. The authors applied their model to urban, suburban and rural scenarios and presented a comparative performance study among the various scenarios in terms of deployment cost, workload assignment, computational resources and energy increase. On the other hand, the authors in [14] introduced an MEC-enabled fiber-wireless (FiWi) access network architecture where computation and storage-centric cloudlets provide reliable cloud services at the edge. A unified cloudlet-aware resource management scheme was proposed where the system is designed in two time division multiple access layers to enhance the network performance and the offloaded traffic is scheduled outside the FiWi transmission slots. To thoroughly study the effectiveness of the proposed solution, a comprehensive analytical framework is developed that is based on different metrics including packet delay, response time, energy efficiency, and battery life.

Other works focused on virtualizing the services and network functions available within an IoT environment. For example, the authors in [15] focused on distributing IoT based applications on edge servers. Specifically, they assigned a unique group of applications to each edge server and defined specific stringent latency requirements for each application. Moreover, they associated each IoT device with multiple edge servers to minimize the total response time. Another work in [16] performed an extensive study to evaluate the performance gains while using mobile edge computing in different scenarios and environments. They specifically considered a game application and studied the effect of device association and server positions on the effectiveness of the deployed solutions. The findings highlighted that the response time greatly depends on the positions of the edge servers and the task assignments between end users and edge servers.

### B. UAV-AIDED IoT NETWORKS

Recently, UAVs have been playing an important role in facilitating the deployment of IoT services. In [17], the authors utilized UAVs for data collection in dense wireless networks where sensors aggregate their data using compressive data

gathering and forward it to designated cluster heads. A UAV then visits all cluster heads along an efficient trajectory towards a remote sink node that processes the data. In [18], the authors deployed UAVs in 3D space to harvest data from IoT devices. They formulated the problem while minimizing the total power and associating different IoT devices to various UAVs. Similarly, in [19] the path of multiple UAVs was optimized to collect data generated by IoT devices efficiently. Initially, they deployed the UAVs within a static network consisting of fixed IoT devices. This step consisted of grouping devices together and associating each group to a single static UAV. Then, they considered the scenario of mobile UAVs where they optimized the path of multiple UAVs to collect IoT data while minimizing the energy consumed.

The research literature related to deploying UAV-mounted cloudlets is relatively limited. Some recent work has tackled the problems associated with using and deploying UAV-mounted cloudlets. For example, in [20], the authors suggested using UAVs as computing cloudlets where they focused on optimizing the allocated bit rates. They carefully designed the path of a single UAV equipped with high computing capabilities while optimizing both uplink and downlink resource allocation. In [21], UAVs were used to support the edge network and act as caching devices. The authors suggested and described a new architecture that can enable effective use of UAVs for both caching and edge computing purposes. The authors in [22] proposed a space-air-ground integrated network that provides edge computing services for different computation-intensive applications. They presented an edge-computing architecture where UAVs provide close edge computing services and satellites provide access to remote cloud computing servers. They first formulated the IoT tasks offloading problem and proposed a reinforcement learning solution. Then, they studied the resource allocation problem on edge servers, by formulating it as a mixed integer program and proposing an efficient heuristic algorithm.

On the other hand, the authors in [23] studied the joint problem of UAV-based computation offloading and trajectory design. Due to its non-convex nature, the authors reformulated the problem to solve it using sequential convex optimization techniques. In [24], a multiple UAV-enabled MEC network was studied. With a fixed number of UAVs, a ground user might offload its computational tasks to the UAV or decide to compute locally. Given the number of computational tasks, the position of the ground users, and the number of available UAVs, the authors studied the sum power minimization problem with latency and coverage constraints. They jointly optimized user associations, power control, computation capacity allocation, and location planning. In order to solve the non-convex problem, the authors proposed an iterative algorithm that gradually solves the three subproblems in an efficient manner.

### C. NOVEL CONTRIBUTIONS

Most of the existing literature on UAV-mounted cloudlets considers either a single UAV or a fixed number of UAVs to

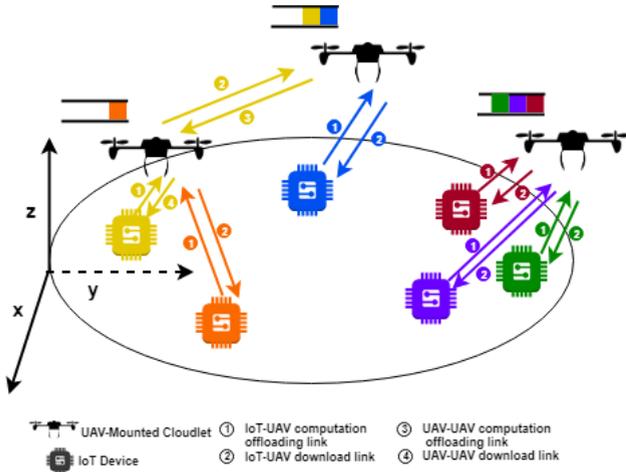
serve a set of devices with special focus on UAV path optimization and energy consumption reduction. In contrast to the related work, this paper presents a novel UAV-enabled edge computing solution for IoT networks by jointly optimizing the number of deployed UAVs, their positions in 3D space, the device-to-UAV associations, and the computation offloading decisions. In particular, we study the optimal 3D deployment of UAV-mounted cloudlets to support latency-sensitive IoT applications while considering a set of constraints related to the computational tasks, available resources, and quality of service targets. The proposed system model enables the efficient utilization of resources while benefiting from the UAVs' unique properties such as flexibility and enhanced line-of-sight connectivity, where each deployed UAV can serve either as an edge computing node or as a relay to other nearby UAVs that are equipped with edge computing capabilities. In order to generate solutions for large scale IoT network scenarios, we also propose an effective meta-heuristic algorithm and demonstrate its low complexity and close-to-optimal performance.

### III. SYSTEM MODEL

We consider an area where  $U$  resource-constrained IoT devices are deployed to regularly monitor and collect information that needs to be processed in a timely manner for proper decision making. Requests to process the collected information are issued to a set of  $D$  UAV-mounted cloudlets, which offer computation offloading services and can serve a maximum of  $K_D$  IoT devices each. The sets of IoT devices and UAV cloudlets are denoted by  $\mathcal{U} = 1, 2, \dots, U$  and  $\mathcal{D} = 1, 2, \dots, D$ , respectively.

Every IoT device is expected to generate computational tasks, whereby requests of multiple devices are delivered to a designated UAV-mounted cloudlet for processing within target latency limits. Due to the low transmission power, some IoT devices may not be able to reach their respective UAV-mounted cloudlets; therefore, additional UAVs may be deployed to serve only as relay nodes without being equipped with edge computing capabilities. Consequently, based on the expected demand, a swarm of UAVs are deployed in 3D space as a fully meshed network. Selected UAVs are equipped with cloudlet resources while others serve as relay nodes to connect IoT devices to their cloudlets. The system model also supports cases where a given UAV can serve as a cloudlet for a set of IoT devices and as a relay for other devices. In this work, we aim at placing the minimum number of UAVs in 3D space to handle all computational workload generated by the IoT devices in the network. Each IoT device is associated to one of the nearby UAVs denoted as *home UAV* that may either relay the tasks to another UAV for computation or compute the tasks locally. The UAV responsible for computing an offloaded task is denoted as the *cloudlet UAV*.

Figure 1 presents a sample network with six IoT devices offloading tasks to three UAV-mounted cloudlets. The device, first, associates with a nearby UAV to act as its home UAV and uploads its computation workload. Afterwards, the home



**FIGURE 1.** Sample network scenario with multiple IoT devices offloading computational tasks to a set of deployed UAV-mounted cloudlets.

UAV may compute the tasks locally and deliver the results to the respective IoT device upon completion or offload to another UAV that can handle the requests. If offloaded to another UAV, the latter computes the task and sends the result back to the home UAV to deliver it to the requesting IoT device. For example in Fig. 1 the IoT device in yellow is connected to one UAV but the offloaded task is computed by another UAV. The IoT device in blue is, however, served by a close UAV that acts as both its home as well as cloudlet UAV.

**A. AIR-TO-GROUND COMMUNICATION CHANNEL**

Accurate channel characterization is essential for the design of efficient UAV communication systems. A lot of work has been done to model air-to-ground channels between UAVs and end devices [25]. In this work, we consider the channel path loss model from [26] since it captures several relevant characteristics and is given as follows:

$$PL_D = 20 \log \left( \frac{4\pi f_c}{c} \right) + 20 \log \left( \sqrt{h^2 + r^2} \right) + p_{LoS} \eta_{LoS} + (1 - p_{LoS}) \eta_{NLoS}, \quad (1)$$

where  $h$  is the height between the UAV and the ground device,  $r$  is the horizontal distance between the device and the UAV,  $f_c$  is the carrier frequency in Hz,  $c$  is the speed of light in m/s,  $\eta_{LoS}$  and  $\eta_{NLoS}$  are the losses corresponding to line-of-sight (LoS) and non-LoS connections, respectively, depending on the environment. The LoS probability can be determined as follows:

$$p_{LoS} = \frac{1}{1 + \mu \cdot \exp \left( -\beta \left( \arctan \left( \frac{h}{r} \right) - \mu \right) \right)}, \quad (2)$$

where  $\mu$  and  $\beta$  are constants that depend on the environment.

**B. IoT COMPUTATION TASKS**

We consider that each IoT device  $i$  generates computation requests based on a Poisson distribution with an average rate  $\lambda_i$  [27]–[29]. All requests generated by the group of

devices associated with a given UAV-mounted cloudlet are aggregated at the cloudlet. Consequently, the arrival process of the aggregate requests to one cloudlet UAV  $j$  also follows a Poisson distribution with an average rate that is equal to the summation of the individual rates at which each IoT device generates tasks. Moreover, we consider computation tasks that need to be completely processed within a given delay limit depending on the type of data collected by the device and the offered application. Each IoT device  $i$  requires its tasks to be fulfilled before a time limit  $T_i$ .

**C. UAV COMPUTING RESOURCES**

Each UAV  $j$  is assumed to execute IoT requests in an exponentially distributed manner with an average service time equals to  $\frac{1}{\mu_j}$ , where  $\mu_j$  is the average service rate of UAV  $j$ . If the processing capacity of UAV  $j$  is  $f_j$  cycles/sec, then the service rate in requests/sec is  $f_j/L$ , where  $L$  is the average size of the computation tasks in cycles. Given a Poisson arrival of requests and an exponential service time distribution, every UAV-mounted cloudlet is modeled as an M/M/1 queuing system with arrival rate  $\sum_i \lambda_i$  and service rate  $f_j/L$  [9], [10], [30]. The summation  $\sum_i \lambda_i$  represents the aggregated rates  $\lambda_i$  of all IoT devices  $i$  assigned to a given UAV-mounted cloudlet.

**D. COMPUTATION TASKS: TOTAL DELAY**

In case IoT device  $i$  offloads its tasks to one of the UAVs, then the incurred delay comprises the time  $t_{up,ij}$  to upload the data to the home UAV  $j$ , the time  $t_{U2U,jk}$  to deliver the task from the home UAV  $j$  to the cloudlet UAV  $k$ , the time  $t_{process,ik}$  to fully process the task at the cloudlet UAV  $k$ , and the time  $t_{down,ij}$  to deliver the result back to the IoT device through the home UAV  $j$ . All UAVs are assumed to be fully meshed and the transfer delay  $t_{U2U,jk}$  of data from one UAV to another is assumed to be fixed. We note that the same UAV may act as both home UAV and cloudlet UAV, hence, in this case  $t_{U2U,jk} = 0$ . As in other related works, such as [10] and [19], the size of the task output data is in general assumed to be much smaller than the size of the task input data uploaded by the IoT device; therefore, the delay to transfer the result from the home UAV to the IoT device in the downlink is relatively small and can be ignored; i.e.,  $t_{down,ij}$  is set to zero. Consequently, the total delay experienced by IoT device  $i$  can be calculated as follows:

$$t_{total,i} = t_{up,ij} + 2 * t_{U2U,jk} + t_{process,ik}, \quad (3)$$

The upload delay  $t_{up,ij}$  depends on the resulting bit rate of the uplink from the IoT device  $i$  to the home UAV  $j$ . For simplicity, we assume that the uplink bandwidth is equally distributed among all active devices  $A_j$  associated with UAV  $j$  resulting in the following bit rate:

$$R_{ij} = \frac{B_j}{A_j} \log(1 + SNR_{ij}), \quad (4)$$

where  $B_j$  is the total uplink bandwidth of UAV  $j$  and  $SNR_{ij}$  is the received signal-to-noise ratio of IoT device  $i$  at UAV  $j$ .

As for the delay  $t_{\text{process},ik}$ , which represents the total delay spent at the UAV-mounted cloudlet including both the waiting time and the execution time, it is computed according to Little's law knowing that the cloudlet is modeled as an M/M/1 system. Therefore,  $t_{\text{process},ik}$  can be derived as follows:

$$t_{\text{process},ik} = \frac{1}{\frac{f_k}{L} - \sum_n \lambda_n}, \quad (5)$$

where  $\sum_n \lambda_n$  is the sum task generation rates of all IoT devices  $n$  assigned to UAV-mounted cloudlet  $k$  for processing their computation tasks.

#### IV. PROBLEM FORMULATION

We mathematically formulate our problem in this section to determine the minimum number of UAVs in addition to optimal positions in 3D space to serve all IoT computation requests before their latency limits. An IoT device establishes network association with only one home UAV and its requested tasks are processed by one cloudlet UAV. The IoT device could either have the same UAV acting as both home UAV and cloudlet UAV or have two distinct UAVs with the respective roles. The need for two distinct UAVs arises whenever an IoT device is in the coverage range of a UAV with inadequate computation capacity to process its tasks. As a result, the device establishes network connectivity with the closer UAV that in turn relays the device's tasks to another more capable UAV to act as the cloudlet UAV. This kind of relaying happens as long as the latency target is not violated. The formulated optimization problem determines the number of UAVs to be deployed, the position of each deployed UAV, the association of IoT devices to UAVs, and the role of each UAV with respect to each IoT device whether serving as a home UAV or a cloudlet UAV. The solution of the formulated problem ensures that the delay constraint of all devices is being satisfied.

To formulate the problem, we introduce a decision variable  $d_j$  to indicate whether UAV  $j$  is deployed or not.

$$d_j = \begin{cases} 1, & \text{if UAV } j \text{ is deployed} \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

Since each IoT device may use up to two UAVs, a home UAV that connects the device to the UAV network and a cloudlet UAV that acts as an edge server and processes the computation tasks, we make use of two binary decision variables,  $a_{ij}$  and  $b_{ij}$ .  $a_{ij}$  specifies whether UAV  $j$  is the home UAV of IoT device  $i$  and is defined as:

$$a_{ij} = \begin{cases} 1, & \text{if IoT device } i \text{ is associated with UAV } j \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

The other decision variable  $b_{ij}$  indicates whether UAV  $j$  is the cloudlet UAV of IoT device  $i$  and is defined as:

$$b_{ij} = \begin{cases} 1, & \text{if tasks of IoT device} \\ & i \text{ are processed by UAV } j \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

For an IoT device  $i$  to be considered connected to UAV  $j$ , the received SNR at the UAV should be above a target threshold  $\Gamma$ . Hence,

$$P_{ij} \geq a_{ij}\sigma^2\Gamma, \quad (9)$$

constitutes the received power constraint for device  $i$  to associate with UAV  $j$ , where  $P_{ij}$  is the received power level that is calculated according to (1) and  $\sigma^2$  is the thermal noise power.

According to the above, the problem is formulated as a mixed integer program as follows:

$$\text{minimize } \sum_{j=1}^D d_j \quad (10)$$

$$\text{subject to } a_{ij} \leq d_j \quad \forall i \in [1, U] \quad \forall j \in [1, D] \quad (11)$$

$$b_{ij} \leq d_j \quad \forall i \in [1, U] \quad \forall j \in [1, D] \quad (12)$$

$$\sum_{j=1}^D a_{ij} = 1 \quad \forall i \in [1, U] \quad (13)$$

$$\sum_{j=1}^D b_{ij} = 1 \quad \forall i \in [1, U] \quad (14)$$

$$P_{ij} \geq a_{ij}\sigma^2\Gamma \quad \forall i \in [1, U] \quad \forall j \in [1, D] \quad (15)$$

$$\sum_{j=1}^D (a_{ij}t_{\text{up},ij} + \frac{b_{ij}}{\frac{f_j}{L} - \sum_{n=1}^U b_{nj}\lambda_n} + \bar{a}_{ij}b_{ij} \sum_{k=1}^D a_{ik}t_{\text{U2U},jk}) \leq T_i \quad \forall i \in [1, U] \quad (16)$$

$$\frac{f_j}{L} - \sum_{i=1}^U b_{ij}\lambda_i \geq 0 \quad \forall j \in [1, D] \quad (17)$$

$$\sum_{i=1}^U a_{ij} \leq K_D \quad \forall j \in [1, D] \quad (18)$$

$$\sum_{j=1}^D \sum_{k=1}^D \sqrt{(x_j - x_k)^2 + (y_j - y_k)^2 + (z_j - z_k)^2} \geq \theta \quad \forall i, j \in [1, D] \quad (19)$$

The formulated problem attempts to optimize the deployment of UAVs while meeting different constraints related to the computation tasks, time limits, computational resources available, and power constraints. The objective function in (10) minimizes the number of deployed UAVs.

The first constraint in (11) ensures that an IoT device  $i$  is associated to a deployed UAV  $j$ . The constraint in (12) ensures that the cloudlet UAV of a device  $i$  is also a deployed UAV  $j$ . The constraints in (13) and (14) enforce each device  $i$  to be assigned only to one home UAV and one cloudlet UAV respectively. It is possible that one UAV acts as both a home UAV and a cloudlet UAV to the same device. Constraint (15) ensures that the received power of IoT device  $i$  at its home UAV  $j$  exceeds a target value; this constraint is particularly responsible of ensuring a minimum quality of service (QoS). The next constraint in (16) limits the total

time experienced by tasks generated by IoT device  $i$  to a maximum value  $T_i$  that resembles the desired deadline of the respective tasks. As demonstrated in (3), this delay constitutes different components including the upload time to the home UAV  $j$ , exchange delay between home UAV  $j$  and cloudlet UAV  $k$ , processing time at the cloudlet UAV  $k$ , and finally the download time. The upload time is calculated based on (4) where  $A_j$  in this case is given by  $\sum_{i=1}^U a_{ij}$ . The second component of (16) determines the total time the task spends at the cloudlet UAV to be fully processed as in (5) while the last component resembles  $d_{U2U,jk}$  that is non-zero whenever the home UAV is different than the cloudlet UAV. Hence, in this case we take the inverse of  $a_{ij}$  ( $\bar{a}_{ij} = 1 - a_{ij}$ ) to make sure that device  $i$  is not served by UAV  $j$  and we include  $b_{ij}$  to ensure that device  $i$  traffic is being processed by UAV  $j$ . We introduce the summation  $\sum_{k=1}^D a_{ik}t_{U2U,jk}$  to capture the delay between the home UAV and the cloudlet UAV. Constraints (15) and (16) can be linearized using standard techniques as detailed in Appendix A.

The constraint in (17) ensures a stable queuing system at the cloudlet UAV where the arrival rate of computing requests does not exceed the service rate of the UAV. Constraint (18) makes sure that each home UAV does not exceed its user capacity  $K_D$ . The final constraint in (19) is crucial to separate the positions of any two deployed UAVs  $j$  and  $k$  by a minimum planning distance  $\theta$ . This last constraint can be linearized using Taylor series with multiple iterations until convergence.

#### A. COMPLEXITY ANALYSIS

We show that the formulated problem is NP-hard by dividing it into two sub-problems. The first sub-problem is the UAV deployment and the second sub-problem is the device-to-UAV association and IoT task assignment. The first sub-problem can be reduced to the capacitated facility location problem that is well known to be NP-hard [31]. This problem considers a number of facilities that have to be deployed in order to serve a number of customers. In our problem, the UAVs can be the facilities and the IoT devices the customers. The second sub-problem can be reduced to the assignment problem that is also well known to be NP-hard [32]. This problem requires the distribution of specific objects to different bins. In our problem, the IoT devices and their tasks can be the objects and the home UAVs and the cloudlet UAVs the bins. Hence, it is very hard to obtain an optimal solution to our problem especially for large-scale IoT network scenarios. Thus, we propose a scalable solution approach based on a meta-heuristic optimization as described in the next section.

### V. PROPOSED ALGORITHM BASED ON IONS MOTION OPTIMIZATION

Even though this work does not address real-time network operation that requires prompt reaction to any traffic demand deviation, the high complexity of the problem and the difficulty of obtaining an optimal solution in a reasonable time raise the need for a more efficient solution. The complexity

of the problem is further increased with a growing number of IoT devices lending the optimal solution unattainable. In this section, an efficient meta-heuristic algorithm is proposed to provide an optimized solution for large-scale networks in reasonable time. The proposed algorithm can thus be applied in response to any considerable change in traffic demand including IoT task generation rate, task latency targets, devices' positions, and number of devices.

The proposed meta-heuristic algorithm is based on ions motion optimization that iteratively searches for an efficient solution that encompasses the number and positions of UAVs in addition to the device associations and computation task allocations. Ions motion optimization (IMO) was first suggested in [7] and is derived from the natural behavior of ions within different environments. It is a population based meta-heuristic where candidate solutions are represented as anions and cations. Anions are ions assigned negative charges and cations are ions assigned positive charges. Half of the population is considered as anions and the other half is considered as cations. When all these ions exist in the same environment, they form an electrically charged field where unlike ions attract and likewise ions repel. So, in an electrically charged environment, anions move towards cations and cations move towards anions. However, the behaviour of these ions and particles depend on the states of matters in which these particles are formed.

We customize the IMO approach to provide a scalable and efficient solution to our formulated optimization problem. To the best of our knowledge, this is the first work to apply IMO to solve an emerging problem in wireless network design. In the proposed algorithm, we consider two states according to [7], a liquid state where ions have greater freedom of motion followed by a solid state where ions are constrained to small-scale vibrations in their positions. In addition to these two states, the execution of the algorithm starts with an initialization phase and ends with a termination phase as described in the next subsection.

The main motivation behind selecting this solution approach is its fast convergence, ability to try to avoid local optima, and relatively small number of parameters that need be tuned compared to other meta-heuristic algorithms. In [7], authors showed that ion motion optimization provides better conversion behavior as compared to other meta-heuristics including genetic algorithms and particle swarm optimization. The algorithm's fast convergence can be attributed to the liquid state where ions are attracted to the best anion and cation so promising search spaces are explored. Also, the algorithm avoids local optima due to the solid state where ions are forced to exit from the local optimum due to the repulsion forces applied. The balance between exploration and exploitation impacts the positioning of different UAVs and their roles and contributes to the fast conversion.

#### A. IMO INITIALIZATION PHASE

Considering the three variables  $d_j$ ,  $a_{ij}$ , and  $b_{ij}$ , we initialize our population of candidate solutions and divide it equally

into negative charged ions named anions and denoted as AI, and positive charged ions named cations and denoted as CI. Each ion represents a single candidate solution composed of the following: a matrix  $\mathcal{P}$  representing the positions of the UAVs; an association matrix  $\mathcal{A}$  representing which device is connected to which UAV being its home UAV; a task allocation matrix  $\mathcal{B}$  representing which device is served by which UAV being its cloudlet UAV.

The population is initialized as follows:

$$AI_{sv} = \text{Min}_v + (\text{Max}_v - \text{Min}_v)r_1 \quad (20)$$

$$CI_{sv} = \text{Min}_v + (\text{Max}_v - \text{Min}_v)r_2 \quad (21)$$

where the index  $s$  represents the ion number (candidate solution number) and the index  $v$  represents the dimension. The dimension is used to index the different variables in a single solution, where the variables are the UAV positions  $\mathcal{P}$  and the two association matrices  $\mathcal{A}$  and  $\mathcal{B}$ . The upper bounds of the different variables are expressed by Max and the lower bounds are expressed by Min. The variables  $r_1$  and  $r_2$  are arbitrary numbers that are randomly chosen between 0 and 1. The fitness value of each candidate solution depends on the total number of deployed UAVs, the total number of cloudlet UAVs and IoT devices that do not satisfy the problem constraints represented in (11) till (19). The fitness value is calculated as follows:

$$\sum_{j=1}^D d_j + \sum_{j=1}^D q_j + \sum_{i=1}^U k_i, \quad (22)$$

where  $d_j$  indicates if UAV  $j$  is deployed,  $q_j$  indicates if UAV  $j$  violates any of the relevant problem constraints and similarly  $k_i$  indicates whether device  $i$  violates any of the problem constraints. Hence, a lower fitness value indicates a better candidate solution with lower number of deployed UAVs and less number of unsatisfied constraints. The associations between the IoT devices and home UAVs are calculated according to (23) for anions and similarly for cations where  $a_{ij}$  represents whether IoT device  $i$  is associated to UAV  $j$ . To infer the association between the IoT devices and cloudlet UAVs, the same calculations are done for  $b_{ij}$  representing whether device  $i$  is served by cloudlet UAV  $j$ .

$$a_{ij} = \lfloor AI_{sv} \rfloor \bmod 2. \quad (23)$$

### B. IMO LIQUID PHASE

In the liquid phase, ions are attracted to better ions of opposite charges. Cations will move towards the best anions and anions will move towards the best cations. Repulsion forces are neglected in this phase in order to explore the full search space. This behavior is represented on the sample schematic shown in Figure 2 left.

We consider the distance as the key factor affecting the force between opposite ions; thus, the force  $FA_{sv}$  applied on anion  $s$  is calculated as follows [7]:

$$FA_{sv} = \frac{1}{1 + e^{\frac{-0.1}{DA_{sv}}}}, \quad (24)$$

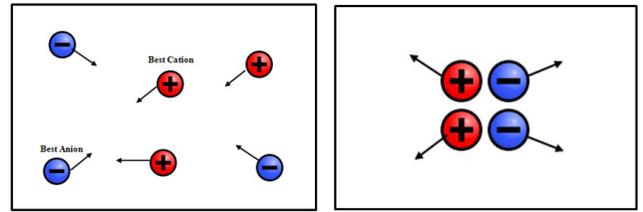


FIGURE 2. The figure on the left shows the behaviour of ions in the liquid phase (more freedom to move in the search space) and the one on the right shows the behavior in the solid phase (local vibration with limited freedom to move).

where  $DA_{sv}$  is the distance between anion  $s$  and the best cation in dimension  $v$  computed as  $|AI_{sv} - \text{BestC}_v|$ .  $\text{BestC}_v$  represents the best cation in dimension  $v$ , which is the one with the least fitness value. The forces on cations are calculated in a similar manner.

After calculating the forces applied on each ion, the position of each anion is updated according to (25) and similarly the position of each cation is updated.

$$AI_{sv} = AI_{sv} + FA_{sv}(\text{BestC}_v - AI_{sv}). \quad (25)$$

This process is executed over multiple iterations causing more ions to interact and get attracted to the best ions (anions or cations) with opposite charges. With increasing the number of iterations, ions converge toward one solution in the search space and hence enters the solid phase.

### C. IMO SOLID PHASE

After the ions converge to an optimized solution causing them to enter the solid phase, anions and cations are trapped and have very limited freedom of motion as illustrated in the right of Figure 2. This convergence, however, might have led to a local optimum and thus requires an external force to crack the solid structure causing ions to move away from each other in random directions. This mechanism of exiting from a local optimum is described in Algorithm 1 where  $\phi_1$  and  $\phi_2$  take arbitrary values between  $-1$  and  $1$  and  $\text{random}()$  is a function that returns a random number between 0 and 1.  $\text{BestCfit}$  is the cation with the best fitness value and  $\text{BestAfit}$  is the anion with the best fitness value. On the other hand,  $\text{WorstCfit}$  represents the cation with the worst fitness value and  $\text{WorstAfit}$  is the anion with the worst fitness value.

### D. IMO TERMINATION PHASE

The algorithm keeps on iterating between the solid and liquid states until the maximum number of iterations is reached and, thus, the best feasible candidate solution is returned. Depending on the nature of the problem formulation and the number of iterations, the outcome can coverage to the global optimum or can lead to the best achieved local optimum.

## VI. SIMULATION RESULTS AND PERFORMANCE ANALYSIS

In this section, we evaluate the proposed meta-heuristic solution approach for the problem formulated in Section IV.

**Algorithm 1** IMO Solid Phase Implementation Based on [7]

```

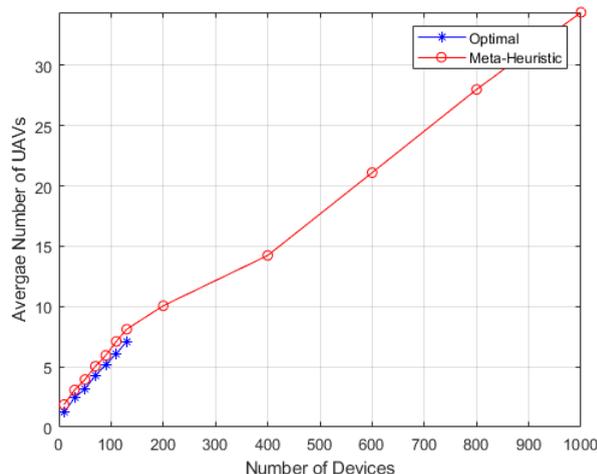
0: procedure Solid-Phase
0:   if (BestCfit  $\geq \frac{\text{WorstCfit}}{2}$  and BestAfit  $\geq \frac{\text{WorstAfit}}{2}$  )
0:     then
0:       if (random()  $\geq 0.5$  ) then
0:          $AI_s = AI_s + \phi_1(\text{BestC} - 1)$ 
0:       else
0:          $AI_s = AI_s + \phi_1(\text{BestC})$ 
0:       end if
0:       if (random()  $\geq 0.5$  ) then
0:          $CI_s = CI_s + \phi_2(\text{BestA} - 1)$ 
0:       else
0:          $CI_s = CI_s + \phi_2(\text{BestA})$ 
0:       end if
0:       if (random()  $\leq 0.05$  ) then
0:         Randomly re-initialize all  $AI_s$  and  $CI_s$ 
0:       end if
0:     end if
0:   end procedure
    
```

We generate extensive simulation results to demonstrate its effectiveness in terms of complexity and performance with respect to the optimal solution. We present results as a function of various system parameters and for multiple IoT use cases with different latency requirements. We note that the formulated optimization problem is solved using the mixed integer linear programming solver of MATLAB’s optimization toolbox after being linearized as detailed earlier; the IMO algorithm is implemented using JAVA programming language.

We consider a 200 m × 200 m area where IoT devices are randomly deployed and consider the system parameters summarized in Table 1 unless otherwise specified. First, we start by evaluating the gap between the optimal solution and the proposed meta-heuristic approach for various scenarios. Fig. 3 shows the average number of deployed UAVs with respect to the number of IoT devices. We can see that the meta-heuristic algorithm provides close-to-optimal results. Minimizing the number of UAVs deployed in an IoT network leads to lower costs and better utilization of available resources. Increasing the number of IoT devices leads to a larger network scenario and exponentially impacts

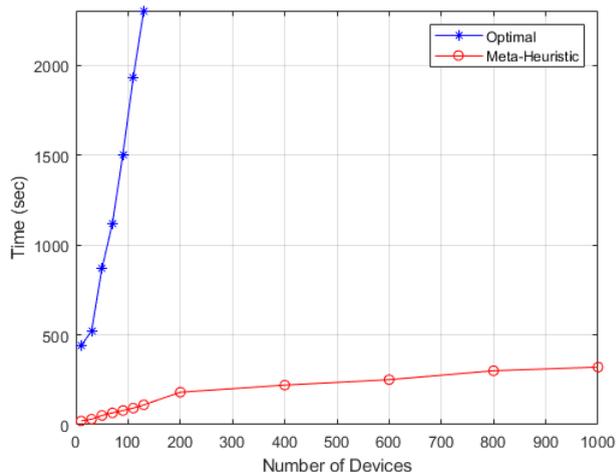
**TABLE 1.** System parameters used in the simulations.

Parameter	Value
UAV transmit power $P_D$	5 Watts
UAV maximum capacity $K_D$	30
Thermal noise power $\sigma^2$	$10^{-6}$ Watts
SNR threshold $\Gamma$	2 dB
carrier frequency $f_c$	2.5 GHz
$\mu$ in (2)	9.61
$\beta$ in (2)	0.16
$\eta_{LoS}$	1
$\eta_{NLoS}$	20
Cloudlet processing capacity $f_j$	7 GHz



**FIGURE 3.** Average number of UAVs required to serve the IoT devices for the optimal solution compared to the proposed meta-heuristic algorithm, as a function of the total number of IoT devices.

the execution time to generate the optimal solution as demonstrated in Fig. 4. The meta-heuristic approach, on the other hand, is shown to be a very efficient alternative to generate near-optimal results with low complexity for relatively large networks.



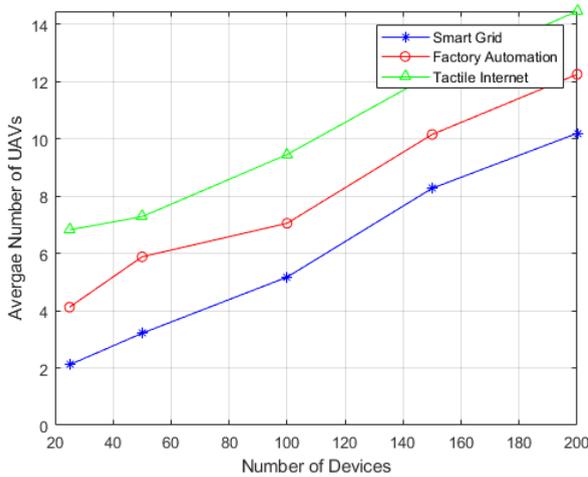
**FIGURE 4.** Execution time of the optimal solution versus the meta-heuristic approach as a function of the number of IoT devices.

Next, we deploy different applications on the IoT devices and consider various industry verticals. Table 2 presents the deadline ranges of the considered industry verticals and the maximum latency limit that we applied in our simulations.

**TABLE 2.** IoT industry verticals with their deadline ranges as per [33]. The last column presents the selected deadline in the simulations.

Industry Vertical	Deadline Range	Applied Latency Limit
Smart Grid	3 – 20 ms	20 ms
Factory Automation	0.25 – 10 ms	10 ms
Tactile Internet	1 – 10 ms	5 ms

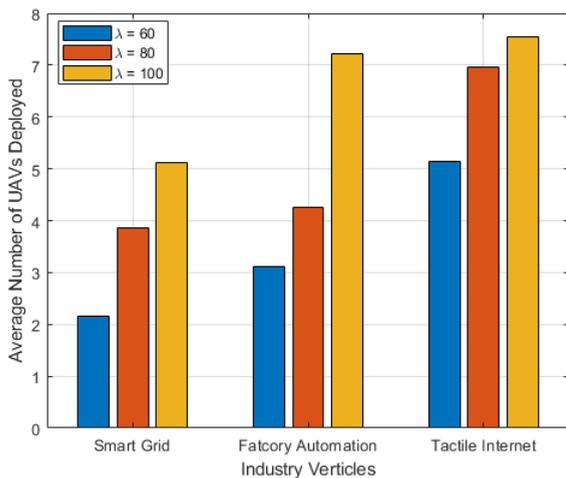
In Fig. 5, we plot the average number of deployed UAVs as a function of the number of IoT devices available in the network with  $\lambda$  set to 90 requests/sec. We compare the three



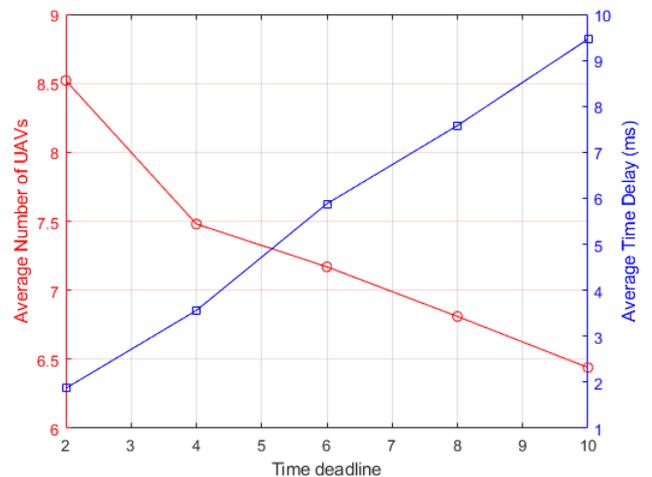
**FIGURE 5.** Average number of UAVs required to serve the IoT devices based on the proposed meta-heuristic approach as a function of the total number of IoT devices for different industry verticals.

different IoT industry verticals while taking into account their latency limits. We clearly infer that as the number of devices increases causing an increase in the number of requests, more UAVs are needed to serve all tasks. The number of utilized UAVs is highest for the Tactile Internet use case due to its stringent latency requirements.

In Fig.6, we study the average number of UAVs deployed for different verticals and according to different request rates; the computation task request rate  $\lambda$  is varied between 60 and 100 requests/sec. The figure shows that as the request rate increases the number of deployed UAVs also increases. Moreover, as the time deadline of the different verticals gets stricter, the number of deployed UAVs increases to process all generated workload in shorter time. For example, for a  $\lambda$  of 60 requests/sec the smart grid use case requires the deployment of around 2.25 UAVs on average whereas the factory automation use case requires the deployment of 3.25 UAVs on average to serve the same number of IoT devices.



**FIGURE 6.** Average number of UAVs required to serve IoT devices in different industry verticals as a function of the rate of computation task requests.



**FIGURE 7.** Average number of UAVs and achieved time delay as a function of the time deadline for the Tactile Internet use case.

To further study the effects of time deadlines and response times, we plot in Fig.7 the average number of UAVs deployed to serve 200 users with  $\lambda$  set to 90 requests/sec. In addition, the figure presents the actual achieved time delay as a function of the time deadline. We vary the required time deadline between 1 and 10 ms and show that the number of UAVs can be dropped from 8.5 to around 2.2 on average for the same number of IoT devices and same level of computation workload. We notice that the actual achieved response time is very close to the applied deadline; this is due to the fact that our solution aims at minimizing the resources in terms of the number of UAVs.

**VII. CONCLUSION**

In this work, we leveraged the flexibility, enhanced line of sight connectivity, and low cost of UAVs to deploy them as mobile edge computing cloudlets to serve resource-constrained IoT devices. Specifically, we addressed the joint problem of 3D deployment of UAV-mounted cloudlets, IoT device to UAV association, and IoT device to UAV-mounted cloudlet computation task offloading with focus on latency sensitive services. We formulated the problem as a mixed integer program, demonstrated that it is NP-hard, and proposed a low complexity efficient algorithm based on ions motion meta-heuristic optimization. Results are presented and analyzed as a function of a wide range of parameters and quality of service constraints. We demonstrated the effectiveness of the proposed algorithm in terms of achieving close-to-optimal performance with notably low complexity, making it especially attractive for large IoT network scenarios.

**APPENDIX  
LINEARIZATION OF PROBLEM CONSTRAINTS**

In this appendix, we describe the linearization of (15) and (16) using methods from [34].

- 1) The optimization problem constraint (15) can be linearized as follows.

$$\frac{1}{P_{i,j}} \leq a_{i,j} \frac{1}{\sigma^2 \Gamma}$$

$$\frac{1}{P_{i,j}} = \frac{d^2}{\kappa P_T}$$

$$d^2 = (x_d - x_u)^2 + (y_d - y_u)^2 = x_d^2 + x_u^2 - 2x_d x_u + y_d^2 + y_u^2 - 2y_d y_u$$

$x_d^2$ , and similarly  $y_d^2$ , can be linearized using piecewise linear approximation as per [34]. To do so, consider  $x_d$  is limited between  $X_2$  as a lower bound and  $X_1$  as an upper bound; thus  $X_2 \leq x_d \leq X_1$ . The piecewise linear approximation of  $x_d^2$  can then be written as:

$$\begin{aligned} \lambda_1(X_2)^2 + \lambda_2(X_1 \frac{1}{3})^2 + \lambda_3(X_1 \frac{2}{3})^2 + \lambda_4(X_1)^2 &= x_d^2 \\ \lambda_1(X_2) + \lambda_2(X_1 \frac{1}{3}) + \lambda_3(X_1 \frac{2}{3}) + \lambda_4(X_1) &= x_d \\ \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 &= 1 \end{aligned}$$

No other constraints are needed on  $\lambda$  since  $x^2$  is convex.

- 2) The component  $\bar{a}_{ij} b_{ij} \sum_{k=1}^D a_{ik} t_{U2U,jk}$  of the problem constraint represented in (16) can be linearized as follows. We first consider  $\bar{a}_{ij} b_{ij}$ , being a product of two binary variables, and replace it with another binary variable  $z_{ij} \in \{0, 1\}$  conditioned by the following constraints to force it to take the value of  $\bar{a}_{ij} b_{ij}$ .

$$z_{ij} \leq \bar{a}_{ij}$$

$$z_{ij} \leq b_{ij}$$

$$z_{ij} \geq \bar{a}_{ij} + b_{ij} - 1$$

Next, the component  $\bar{a}_{ij} b_{ij} \sum_{k=1}^D a_{ik} t_{U2U,jk}$  becomes  $z_{ij} \sum_{k=1}^D a_{ik} t_{U2U,jk}$ ; this involves product of two binary variables as above since both of  $z_{ij}$  and  $a_{ik}$  are binary while  $t_{U2U,jk}$  is a given parameter. We replace  $z_{ij} \sum_{k=1}^D a_{ik} t_{U2U,jk}$  by a new variable  $s_{ij}$  and consider the maximum value of all  $t_{U2U,jk}$ ,  $\forall j, \forall k$  as  $T_{U2U,jk}$ . The following constraints are then added to force  $s_{ij}$  to take the value of  $z_{ij} \sum_{k=1}^D a_{ik} t_{U2U,jk}$ :

$$s_{ij} \leq T_{U2U,jk} z_{ij}$$

$$s_{ij} \leq \sum_{k=1}^D a_{ik} t_{U2U,jk}$$

$$s_{ij} \geq \sum_{k=1}^D a_{ik} t_{U2U,jk} - T_{U2U,jk} (1 - z_{ij})$$

$$s_{ij} \geq 0$$

- 3) The component  $\frac{b_{ij}}{\mu_j - \sum_{n=1}^U b_{nj} \lambda_n}$  of the problem constraint represented in (16) represents a product of a binary variable  $b_{ij}$  and a continuous variable  $m_j = \frac{1}{\mu_j - \sum_{n=1}^U b_{nj} \lambda_n}$ . The latter product is denoted by  $h_{ij} = \frac{1}{\mu_j - \sum_{n=1}^U b_{nj} \lambda_n}$ .

We can linearize  $b_{ij} m_j$  using the following constraints, where  $M_j$  is an upper bound on  $m_j$ .

$$h_{ij} \leq M_j b_{ij}$$

$$h_{ij} \leq m_j$$

$$h_{ij} \geq m_j - M_j (1 - b_{ij})$$

$$h_{ij} \geq 0$$

Then, we deal with  $m_j = \frac{1}{\mu_j - \sum_{n=1}^U b_{nj} \lambda_n}$ . It becomes  $m_j \mu_j - m_j \sum_{n=1}^U b_{nj} \lambda_n = 1$ . The component  $m_j \sum_{n=1}^U b_{nj} \lambda_n$  comprises a product of continuous and binary variables and can also be linearized as above.

## REFERENCES

- [1] G. Premsankar, M. Di Francesco, and T. Taleb, "Edge computing for the Internet of Things: A case study," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1275–1284, Apr. 2018.
- [2] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, "A vision of IoT: Applications, challenges, and opportunities with china perspective," *IEEE Internet Things J.*, vol. 1, no. 4, pp. 349–359, Aug. 2014.
- [3] L. Tong, Y. Li, and W. Gao, "A hierarchical edge cloud architecture for mobile computing," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2016, pp. 1–9.
- [4] X. Sun and N. Ansari, "EdgeIoT: Mobile edge computing for the Internet of Things," *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 22–29, Dec. 2016.
- [5] W. Fawaz, R. Atallah, C. Assi, and M. Khabbaz, "Unmanned aerial vehicles as store-carry-forward nodes for vehicular networks," *IEEE Access*, vol. 5, pp. 23710–23718, 2017.
- [6] M. Samir, S. Sharafeddine, C. Assi, T. Nguyen, and A. Ghreyeb, "UAV trajectory planning for data collection from time-constrained IoT devices," *IEEE Trans. Wireless Commun.*, to be published.
- [7] B. Javidy, A. Hatamlou, and S. Mirjalili, "Ions motion algorithm for solving optimization problems," *Appl. Soft Comput.*, vol. 32, pp. 72–79, Jul. 2015.
- [8] H. A. Alameddine, S. Sharafeddine, S. Sebbah, S. Ayoubi, and C. M. Assi, "Dynamic task offloading and scheduling for low-latency IoT services in multi-access edge computing," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 668–682, Mar. 2019.
- [9] N. Kherraf, H. A. Alameddine, S. Sharafeddine, C. M. Assi, and A. Ghreyeb, "Optimized provisioning of edge computing resources with heterogeneous workload in IoT networks," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 2, pp. 459–474, Jun. 2019.
- [10] X. Sun and N. Ansari, "Latency aware workload offloading in the cloudlet network," *IEEE Commun. Lett.*, vol. 21, no. 7, pp. 1481–1484, Jul. 2017.
- [11] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 856–868, Jan. 2019.
- [12] A. Ceselli, M. Premoli, and S. Secci, "Mobile edge cloud network design optimization," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1818–1831, Jun. 2017.
- [13] S. Mondal, G. Das, and E. Wong, "CCOMPASSION: A hybrid cloudlet placement framework over passive optical access networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2018, pp. 216–224.
- [14] B. P. Rimal, D. P. Van, and M. Maier, "Cloudlet enhanced fiber-wireless access networks for mobile-edge computing," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3601–3618, Jun. 2017.
- [15] M. Jia, W. Liang, and Z. Xu, "QoS-aware task offloading in distributed cloudlets with virtual network function services," in *Proc. 20th ACM Int. Conf. Modeling, Anal. Simulation Wireless Mobile Syst.*, 2017, pp. 109–116.
- [16] P. J. Braun, S. Pandi, R.-S. Schmolli, and F. H. P. Fitzek, "On the study and deployment of mobile edge cloud for tactile Internet using a 5G gaming application," in *Proc. 14th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2017, pp. 154–159.
- [17] D. Ebrahimi, S. Sharafeddine, P.-H. Ho, and C. Assi, "UAV-aided projection-based compressive data gathering in wireless sensor networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1893–1905, Apr. 2019.

- [18] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile unmanned aerial vehicles (UAVs) for energy-efficient Internet of Things communications," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7574–7589, Nov. 2017.
- [19] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile Internet of Things: Can UAVs provide an energy-efficient mobile architecture?" in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.
- [20] S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2049–2063, Mar. 2018.
- [21] N. Cheng, W. Xu, W. Shi, Y. Zhou, N. Lu, H. Zhou, and X. Shen, "Air-ground integrated mobile edge networks: Architecture, challenges and opportunities," 2018, *arXiv:1804.04763*. [Online]. Available: <https://arxiv.org/abs/1804.04763>
- [22] N. Cheng, F. Lyu, W. Quan, C. Zhou, H. He, W. Shi, and X. Shen, "Space/aerial-assisted computing offloading for IoT applications: A learning-based approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1117–1129, May 2019.
- [23] F. Zhou, Y. Wu, H. Sun, and Z. Chu, "UAV-enabled mobile edge computing: Offloading optimization and trajectory design," 2018, *arXiv:1802.03906*. [Online]. Available: <https://arxiv.org/abs/1802.03906>
- [24] Z. Yang, C. Pan, K. Wang, and M. Shikh-Bahaei, "Energy efficient resource allocation in UAV-enabled mobile edge computing networks," 2019, *arXiv:1902.03158*. [Online]. Available: <https://arxiv.org/abs/1902.03158>
- [25] A. A. Khuwaja, Y. Chen, N. Zhao, M.-S. Alouini, and P. Dobbins, "A survey of channel modeling for UAV communications," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2804–2821, 4th Quart., 2018.
- [26] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Dec. 2014.
- [27] A. Yousefpour, G. Ishigaki, R. Gour, and J. P. Jue, "On reducing IoT service delay via fog offloading," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 998–1010, Apr. 2018.
- [28] H. S. Narman, M. S. Hossain, M. Atiqzaman, and H. Shen, "Scheduling Internet of Things applications in cloud computing," *Ann. Telecommun.*, vol. 72, nos. 1–2, pp. 79–93, 2017.
- [29] T. G. Rodrigues, K. Suto, H. Nishiyama, and N. Kato, "Hybrid method for minimizing service delay in edge cloud computing through VM migration and transmission power control," *IEEE Trans. Comput.*, vol. 66, no. 5, pp. 810–819, May 2017.
- [30] D. Zeng, L. Gu, S. Guo, Z. Cheng, and S. Yu, "Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system," *IEEE Trans. Comput.*, vol. 65, no. 12, pp. 3702–3712, Dec. 2016.
- [31] L.-Y. Wu, X.-S. Zhang, and J.-L. Zhang, "Capacitated facility location problem with general setup cost," *Comput. Oper. Res.*, vol. 33, no. 5, pp. 1226–1241, 2006.
- [32] S. O. Krumke and C. Thielen, "The generalized assignment problem with minimum quantities," *Eur. J. Oper. Res.*, vol. 228, no. 1, pp. 46–55, 2013.
- [33] P. Schulz, M. Matthe, H. Klessig, M. Simsek, G. Fettweis, J. Ansari, S. A. Ashraf, B. Almeroth, J. Voigt, and I. Riedel, "Latency critical IoT applications in 5G: Perspective on the design of radio interface and network architecture," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 70–78, Feb. 2017.
- [34] *AIMMS Optimization Modeling*, AIMMS, New Delhi, India, 2019.



**RANIA ISLAMBOULI** received the B.S. and M.S. degrees in computer science from the Lebanese American University (LAU), in 2017 and 2019, respectively. She is currently pursuing the Ph.D. degree with the Ecole Polytechnique Federale de Lausanne (EPFL). Her research interests include wireless networking, cloud computing, and distributed systems.



**SANAA SHARAFEDDINE** received the B.E. and M.E. degrees in computer and communications engineering from the American University of Beirut (AUB), in 1999 and 2001, respectively, and the Ph.D. degree in communications engineering from the Munich University of Technology (TUM), in 2005 in collaboration with Siemens AG research labs in Munich. She is currently an Associate Professor of computer science with Lebanese American University. Her research interests include UAV-assisted solutions in 5G and beyond, wireless networking, and multimedia services. She is an Associate Editor of the *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, the Area Editor of Elsevier *Ad Hoc Networks Journal*, and an Associate Editor of *IEEE ACCESS*.

• • •