Energy-Efficient Computation Offloading for UAV-Assisted MEC: A Two-Stage Optimization Scheme

WEIWEI LIN, TIANSHENG HUANG, XIN LI, FANG SHI, and XIUMIN WANG, School of Computer Science and Engineering, South China University of Technology, China
CHING-HSIEN HSU, Department of Computer Science and Information Engineering, Asia University and Department of Medical Research, China Medical University Hospital, China Medical University

In addition to the stationary **mobile edge computing (MEC)** servers, a few MEC surrogates that possess a certain mobility and computation capacity, e.g., flying **unmanned aerial vehicles (UAVs)** and private vehicles, have risen as powerful counterparts for service provision. In this article, we design a two-stage online scheduling scheme, targeting computation offloading in a UAV-assisted MEC system. On our stage-one formulation, an online scheduling framework is proposed for dynamic adjustment of mobile users' CPU frequency and their transmission power, aiming at producing a socially beneficial solution to users. But the major impediment during our investigation lies in that users might not unconditionally follow the scheduling decision released by servers as a result of their individual rationality. In this regard, we formulate each step of online scheduling on stage one into a non-cooperative game with potential competition over the limited radio resource. As a solution, a centralized online scheduling algorithm, called ONCCO, is proposed, which significantly promotes social benefit on the basis of the users' individual rationality. On our stage-two formulation, we are working towards the optimization of UAV computation resource provision, aiming at minimizing the energy consumption of UAVs during such a process, and correspondingly, another algorithm, called WS-UAV, is given as a solution. Finally, extensive experiments via numerical simulation are conducted for an evaluation purpose, by which we show that our proposed algorithms achieve satisfying performance enhancement in terms of energy conservation and sustainable service provision.

CCS Concepts: • **Networks** → **Network management**; • **Mathematics of computing** → *Stochastic processes*; • **Computing methodologies** → *Parallel computing methodologies;*

Additional Key Words and Phrases: Computation offloading, mobile edge computing, nash equilibrium, non-cooperative game, unmanned aerial vehicles

4

This work is supported by National Natural Science Foundation of China (62072187, 61872084), Guangdong Major Project of Basic and Applied Basic Research (2019B030302002), Guangzhou Science and Technology Program key projects (202007040002), Guangzhou Development Zone Science and Technology (2020GH10).
Authors' addresses: W. Lin, T. Huang (corresponding author), X. Li, F. Shi, and X. Wang, School of Computer Science and Engineering, South China University of Technology, Guangzhou, China; emails: linww@scut.edu.cn, cs_tianshenghuang@mail.scut.edu.cn, lixin_forget@163.com, csshifang@mail.scut.edu.cn, xmwang@scut.edu.cn; C.-H. Hsu (corresponding author), Department of Computer Science and Information Engineering, Asia University, Taiwan, Department of Medical Research, China Medical University Hospital, China Medical University, Taiwan; email: robertchh@gmail.com.

1533-5399/2021/10-ART4 $15.00
https://doi.org/10.1145/3430503

ACM Reference format:
Weiwei Lin, Tiansheng Huang, Xin Li, Fang Shi, Xiumin Wang, and Ching-Hsien Hsu. 2021. Energy-Efficient Computation Offloading for UAV-Assisted MEC: A Two-Stage Optimization Scheme. *ACM Trans. Internet Technol.* 22, 1, Article 4 (October 2021), 23 pages.
https://doi.org/10.1145/3430503

1 INTRODUCTION

1.1 Background

With the increasingly ubiquitous usage of intelligent mobile terminals and an explosive surge of the applications' computational demand, minimizing transmission energy consumption and service latency for the **Mobile Users (MUs)** has become an even more urgent call. This particular demand prompts the emergence of **Mobile Edge Computing (MEC)**. Technically, MEC is a network architecture that enables offloading of computationally expensive tasks to the cellular edge. The basic functionality behind MEC is that, by running applications and computing offloading tasks closer to the end devices, network congestion of the backbone network could be nicely reduced and compute-intensive applications could gain a quicker response. By computation offloading, MEC enables MUs, such as a mobile phone or a laptop, to fully or partially transfer its resource-intensive computational tasks to an external cellular platform, and therefore help improve the application computation capability in a cost-effective and energy-saving manner. Among various surrogates serving as MEC servers, **unmanned aerial vehicles (UAVs)** have received notable attention in recent years due to their attractive merits. With their high degree of mobility and lightweight features, UAVs are capable of consistently providing reliable services to potential users, regardless of the geographical confinement and deployed environment. Besides, thanks to their high mobility, UAVs allow us to do real-time scheduling on the mounted computation resources via trajectory planning. In comparison, traditional stationary MEC servers, which are not capable of instantly transferring their computation resources from place to place, would possibly cause an imbalance between resource deployment and actual demand, regarding the high variation of demands in terms of both temporal and spatial scales (try to think about a traffic hotspot in a concert).

1.2 Motivations

Barring the potential advantage of UAV-assisted MEC deployment, we notice that there are a few technical problems that are worth further investigation. Our concern is mainly raised from a resource competition perspective. Considering a case that multiple mobile users doing the transmission at the same time. Then inevitably there may exist interference between users, so the transmission data rate of all the involved users would suffer an undesirable loss due to the mutual impact. Interference (or noise) without proper management could become really devastating and will dramatically degrade the service experience, which is the last thing that service providers want to see. This motivates us to investigate a regulation rule to coordinate the offloading process of the MUs. But a technically workable scheme is not that easy to find. The main impediment lies in that MUs may not honestly follow the regulation when they see more personal interest could be gained by not following. The selfish nature of the mobile users, which we refer to as *individual rationality*, prompts us to extract a potential solution that guarantees interest optimality for all the users, and meanwhile, maintains social benefits from the perspective of a service provider. This motive drives our research over the first stage of scheduling, in which we study how to issue a scheduling recommendation to all users to ensure simultaneously individual rationality and social benefits. Additionally, another problem we are interested in is that how the UAVs schedule their own computation resources after the scheduling decision has been issued on stage one, with the purpose of sustaining their long-term service performance and minimizing their energy consumption at the same time. This particular problem prompts our investigation on stage-two optimization.

1.3 Contributions and Organization

As we are sufficiently motivated by the two research problems we formerly proposed, now we shall briefly introduce our proposed solutions and summarize our contribution towards the current field of study, as listed in the following:

- We formulate a long-term payoff for MUs, in which their workload queue stability and energy consumption are taken into account. To get rid of the time-coupling effects of the long-term queue stability constraint, we then transform the long-term payoff into an online form, based on which, we formulate a non-cooperative game model and try to manage the MUs' mutual interference during transmission.
- We design a centralized algorithm (called ONCCO) to extract *Nash Equilibrium*, which is essentially one of the strategy profiles that does not violate individual rationality, and at the same time, ensure social benefits.
- On our stage-two formulation, we formally formulate a UAVs' computation resource scheduling problem, using a very similar technique to our stage-one formulation. On the basis of the proposed model, we smoothly derive the optimal solution and a rather simple algorithm, called WS-UAV, is proposed.
- We conduct a numerical experiment to investigate the validity and convergence property of our proposed algorithms and corroborate their effectiveness in terms of energy conservation and long-term performance enhancement.

To the best knowledge of the authors, among works that combine game theory, this is the first attempt to formulate an online payoff for each player. In addition, few studies have made use of a centralized scheme to compute a *Nash Equilibrium* and issue it as a scheduling result. At last, we need to note that our practice of centralized Nash Equilibrium extraction for an online scheduling game has the potential to be extended to a wide range of other modern-day systems, and thus, can be regarded as a substantial contribution to the field.

The remainder of this article is organized as follows: Section 2 gives reference and an introduction to previous relevant works. Section 3 establishes the system model of stage-one scheduling and in which we propose our game-theory-based solution. Section 4 formulates the computation resource scheduling problem for UAVs and exhibits the corresponding solution. Section 5 shows the numerical simulation and results of our proposed schemes. Finally, a conclusion is made with the prospect of future attempts in Section 6.

2 RELATED WORKS

Under the environment of MEC, computation offloading has been extensively studied, serving as one of the key techniques to sustain reliable computation services. In [6], Chen formulated a decentralized computation offloading game for **Mobile Cloud Computing (MCC)**, enabling mobile device users to self-organize into the mutually satisfactory computation offloading decisions. In [23], Xia et al. developed two offloading mechanisms, **Selfish Offloading Mechanism (SOM)** and **Global Offloading Mechanism (GOM)**, for mobile equipment with an Android operating system. When applying their proposed framework, the authors claimed that developers don't need to suffer a drastic change in their development procedure for a partial computation offloading application. In [10], Li formulated an M/G/1 queuing model based non-cooperative offloading game for multiple heterogeneous mobile edge cloud, where every player (user equipment) seeks to minimize its average response time. Likewise, in [7], Chen et al. studied the multi-user computation offloading problem on a multi-channel wireless interference environment and formulated it into a potential game. The authors highlighted that the radio resource competition between mobile users on a wireless channel would cause severe interference to each other, hence, a serious degradation on data rates occurs. This radio resource competition is also the key factor we consider in our formulated offloading game.

With the development of modern-edge cloud computing, energy efficiency has become an increasingly important indicator. A number of researchers considered energy consumption as an indicator in their system model (e.g., [2–5, 13, 20, 21, 24]). Chen et al. [5] combined mobile edge computing with environmental-friendly green computing and formulated the optimization problem into a **Maximal Independent Set (MIS)** problem by jointly considering the task arrival rate from mobile devices and the energy harvested rate of mobile edge cloud. You et al. [24] studied the resource allocation problem on the multi-user mobile-edge offloading system with two different communication schemes (i.e., TDMA and OFDMA) where the goal of allocation is to achieve the minimum weighted sum mobile energy consumption. In [20], Wang et al. integrated the **cloud radio access network (C-RAN)** with MEC and further proposed a unifying framework for the power-performance tradeoff of mobile service providers. In [21], Wang et al. focused on a joint optimization problem of communication and computation resources using DVFS technology. They jointly considered two optimization goals (i.e., energy consumption of smart devices and latency of application execution). Like the above-mentioned work, energy efficiency is also a major concern for our work. We focus on the energy consumption on both UAVs and MUs and formulate them as objectives of our optimization problem.

Lately, the combination of mobile edge computing and UAV has received notable attention. In [9], the UAV-Assisted MEC framework was first proposed, in which UAVs are engaged as moving cloudlets, responsible for computing the tasks from mobile users. During their investigation, Jeong et al. optimized the trajectories of UAVs as well as the bit allocation of offloading data. On the basis of [9], Wu et al. extended in [22] the optimization problem from single UAV to multiple drones and jointly considered the optimization of the user scheduling and association, transmit power as well as the UAVs' trajectories. In [8], Cheng et al. further considered the interference between ground base stations and UAVs where UAVs are not only engaged as cloudlets to process the offloaded data traffic from the static base station but also help provide **line-of-sight (LoS)** links to enhance the transmission rate of mobile users. Most of the related line of research focuses on an offline problem formulation, indicating that the scheduling is only done once, i.e., when the cluster is initialized. In [25], an online formulation is proposed, where Zhang et al. no longer posit in advance a known set of tasks with known values of data size and computation complexity. Instead, they model the task arrival as stochastic arrival of bits of data. Though admitting some loss of generality, we deem such a modeling way of task arrival more practical, since the future offloading tasks are hardly predictable and should not be regarded as deterministic.

Though we regard [25] as a major breakthrough in the field, we are prone to believe that the work could be further improved by lifting the strict assumption that radio resource is sufficient and there is no interference or competition among users. Lifting such an assumption may make the modeling as well as the scheduling process much more challenging, but fortunately enough, game theory might serve as a powerful tool to model the competition over limited resources. For example, in [14–16], Messous et al. make use of various kinds of game formulation to model the offloading problems between UAVs, base station, and edge servers. Their works appear to share a few common points with ours in the sense that they also proposed to use a game theory approach to regulate communication interference. But we regret to find their works too evasive about the communication model. More explicitly, we fail to track down how the interference brought by offloading affects the data rate, at least not demonstrated in a formal sense, although they did claim

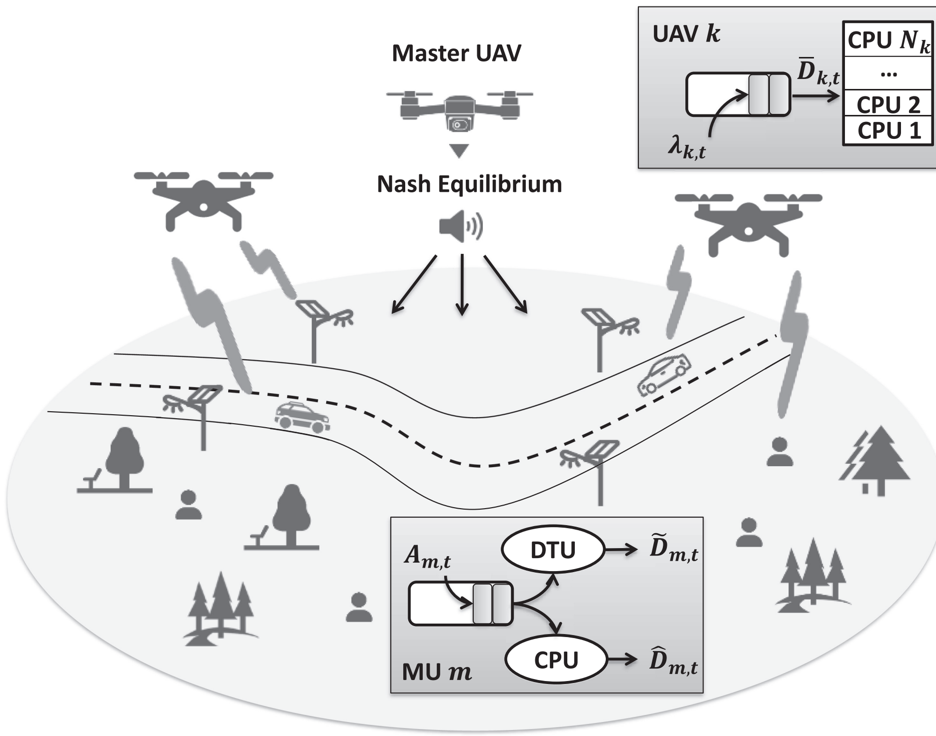


Fig. 1. System model.

that the impact exists. In [26], Zhang et al. similarly proposed a potential game formulation to relieve severe communication and computation resource competition over the offloading process. They make use of an interference-free (or noise-free) model for communication channel (via bandwidth partition), so their model of radio resource competition mainly zeros in on competition of bandwidth, while by contrast, our focus is on the noise management under non-orthogonal multiple access channels. Besides, the works of both Zhang et al. and Messous et al. both adopted a decentralized scheduling process to reach Nash Equilibrium, but we argue that such a decentralized scheme could hardly work in our proposed scenario in the sense that it requires plenty of communication rounds to reach convergence. The lengthy "bargaining" duration is intolerable in our proposed online scheduling scenario since the scheduling is done more frequently (e.g., every 10 s) and correspondingly we must have a higher demand for the scheduling response time. For a more vivid comparison of our works and other former works in the field of UAV involved computation offloading, we refer to Table 1.

3 STAGE ONE: NON-COOPERATIVE OFFLOADING GAME BETWEEN MUS

As shown in Figure 1, the UAV-assisted MEC system consists of M **mobile users (MUs)** and K constantly moving UAVs, captured by $\mathcal{M} = \{1, 2, \ldots, M\}$ and $\mathcal{K} = \{1, 2, \ldots, K\}$, respectively. MUs indexed by m has a workload that is constantly generated by their native applications and they are equipped with limited computing resources[1] that allow local computation for the generated

[1]More explicitly, we assume each MU is equipped with one single CPU.

Table 1. Important Features of State-of-the-art of UAV Involved Computation Offloading

Authors	Type	Objectives	Optimization Targets	Employed techniques	Noise management
X. Gao et al. [1]	offline	• UAV's mission completion time	• UAV trajectory • time allocation for offloading	• Successive Convex Approximation (SCA)	Time Division Multiple Access (TDMA)
F. Zhou et al. [27]	offline	• weighted sum computation bits for users	• UAV trajectory • CPU frequency of users • offloading time • transmit power of users	• SCA • Lagrange duality • subgradient method	TDMA
J. Zhang et al. [25]	online	• average weighted energy consumption of mobile devices and UAVs • task queues' stability for users and UAVs	• UAV trajectory • CPU-cycle frequency of MUs and UAVs • bits of offloading	• Lyapunov optimization • Alternating Direction Method of Multipliers (ADMMs) • interior point method	Frequency Division Multiple Access (FDMA)
M. A. Messous et al. [16]	offline	• delay overhead • UAVs' energy overhead • UAVs' communication cost	• UAVs'offloading decisions, either local computing, offloading to a base station or to a mobile edge server	• a decentrelized algorithm	via potential game
K. Zhang et al. [26]	offline	• MUs' energy consumption • MUs' time latency	• MUs' offloading decisions, either local computing, offloading to a base station or to a UAV	• potential game • a greedy based sequential tuning solution • a decentralized solution that implements a contend process	FDMA
ours	online	• energy consumption of both MUs and UAVs • workload queue's stability for MUs and UAVs	• transmission power of MUs • CPU frequency of MUs and UAVs	• Lyapunov optimization • a centralized scheduling scheme with game theory • convex optimization	via game theory

workload. Also, they are allowed to offload part of the workload to UAVs via **Data Transmission Unit (DTU)** for efficient computation. Here we make an assumption that the workload is measured by bits of data and the workload itself is splittable (i.e., we could send any bits of data for either offloading or computation and there is no correlation between data). This assumption, though with some loss of generality, is typically adopted in the related line of research (see [11], [18], and [28]). In addition, we assume an infinite time horizon, denoted by $\mathcal{T} = \{1, 2, \ldots\}$ for our system and each time slot, with the fixed interval δ, is indexed by t.

3.1 Mobility and Wireless Channel

Now we shall first explain the mobility of UAVs and MUs, which is characterized by their ever-changing geographical position, and then we would quantify the wireless channel gain and the transmission rate according to the relative position and the transmission power.

With the adoption of three-dimensional Euclidean coordinate, we further assume that all the MUs are located on the same horizontal plane and the position of MU m on slot t can be denoted by $[x_{m,t}, y_{m,t}, 0]$. Likewise, we assume the UAVs are all operated in a fixed altitude H and the location of UAV k on interval t is denoted by $[x_{k,t}, y_{k,t}, H]$.

Like [9] and [27], the communication channels between MUs and UAVs are assumed to be dominated by line-of-sight links and the channel power gain between MU m and UAV k is given below:

$$g_{m,k,t} = \frac{g_0}{(x_{m,t} - x_{k,t})^2 + (y_{m,t} - y_{k,t})^2 + H^2} \tag{1}$$

where g_0 represents the channel power gain at the reference distance $d_0 = 1m$ for transmission power of 1 W.

In view of the well-known *Shannon Theory*, the transmission rate of the wireless channel between a pair of MU and UAV can be given as:

$$r_{m,k,t} = B \log_2 \left(1 + \frac{p_{m,k,t} g_{m,k,t}}{\sigma^2 + \sum_{\substack{m'=1 \\ m' \neq m}}^{M} p_{m\prime,k,t} g_{m',k,t}} \right) \tag{2}$$

where $p_{m,k,t}$ represents the transmit power from MU m to UAV k on slot t. B and σ^2 denote bandwidth and additive white Gaussian noise power, respectively. Moreover, we have a maximum cap for the total transmission power of an MU, as given below:

$$\sum_{k \in \mathcal{K}} p_{m,k,t} \leq p_{max}. \tag{3}$$

Given the transmission rate for the coming slot as per (2), we can safely measure the workload consumed by offloading as $\tilde{D}_{m,k,t} = r_{m,k,t}\delta$ and the energy consumption caused by transmission would be $\tilde{E}_{m,k,t} = p_{m,k,t}\delta$. In addition, we derive from the above discussion that given $\{p_{m,k,t}\}_{m \in \mathcal{M}, k \in \mathcal{K}}$ and the relative distance between each pair of MU and UAV, the total consumed workload and the energy consumption for MU m during slot t would be $\tilde{D}_{m,t} = \sum_{k=1}^{K} \tilde{D}_{m,k,t}$ and $\tilde{E}_{m,t} = \sum_{k=1}^{K} \tilde{E}_{m,k,t}$, respectively.

3.2 Local Computing

With the **Dynamic Voltage and Frequency Scaling (DVFS)** technique being employed, we assume that the MUs are able to adjust their local CPU frequency to cope with the ever-changing computation demands. Formally, we let $f_{m,t}$ denote the CPU frequency of MU m during time slot t. Correspondingly, the consumed workload by local computation during slot t can be given as:

$$\hat{D}_{m,t} = \frac{f_{m,t} \cdot \delta}{C} \tag{4}$$

where C denotes the required CPU cycles for computation of 1-bit data. Given the CPU frequency, we can also derive the energy consumption brought by local computation, as follows:

$$\hat{E}_{m,t} = \gamma_m f_{m,t}^3 \delta, \tag{5}$$

where γ_m denotes the effective capacitance coefficient of the MU m.

Besides, we shall make it clear that the CPU frequency could not grow unbounded and due to which we have the following constraint for $f_{m,t}$:

$$0 \leq f_{m,t} \leq f_{max}, \tag{6}$$

where f_{max} is the maximum cap for CPU frequency.

3.3 Workload Arrival and Queuing

For our modeling of workload, it is assumed that the workload will constantly arrive in MUs and the amount of arrival is visible to them at the beginning of a time slot. The arrival amount is featured by a random variable (r.v.) $A_{m,t}$ and is assumed to be drawn from a fixed stochastic distribution, but the exact distribution is not known *a priori* for either the MUs or the UAVs. The arrived workload would be contained in a queue that each MU maintains for future scheduling. The workload queue, with its backlog denoted by $Q_{m,t}$, evolves according to the following rule:

$$Q_{m,t+1} = \max\{Q_{m,t} - D_{m,t}, 0\} + A_{m,t}, \tag{7}$$

where $D_{m,t}$ denotes the amount of workload that is consumed by either workload offloading or local computation. Combining our former discussion on the consumed workload, we have $D_{m,t} = \hat{D}_{m,t} + \sum_{k=1}^{K} \tilde{D}_{m,k,t}$. Note that, in our formulation. it is likely that $D_{m,t} > Q_{m,t-1}$ such that more workload is being scheduled than the existing amount in the queue. For this case, we can simply assume that an extra volume of dummy data was offloaded to UAV or being computed by the local CPU. But we shall clarify later that the redundant scheduling could be well contained with our proposed method, though we admit that we cannot completely rule out its probability to happen if the setting of our algorithm is in an extreme case. In addition, it is assumed that the arrival of workload for each slot could not exceed a constant number and so does the consumed workload. The assumption is w.l.o.g since the data generated by users could not be infinite, and also, the CPU frequency and transmission power should have a maximum cap. Formally, we ensure $A_{m,t} \leq A_{max}$ and $D_{m,t} \leq D_{max}$.

3.4 Long-Term Payoff Function and Its Solution Space

Now we shall formally formulate the payoff function for every MU. In other words, we assume each MU might seek to optimize this particular function over its feasible solution space. Alternatively, we could comprehend it as a scheduling problem for a specific MU m.[2] Formally, we show that:

$$\begin{aligned}
\textbf{(P1)} \quad & \underset{\{p_{m,k,t}\},\{f_{m,t}\}}{minimize} \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[\hat{E}_{m,t} + \sum_{k=1}^{K} \tilde{E}_{m,k,t}\right] \\
s.t. \quad & \sum_{k=1}^{K} p_{m,k,t} \leq p_{max} \qquad t \in \mathcal{T} \\
& p_{m,k,t} \geq 0 \qquad \forall k \in \mathcal{K}, t \in \mathcal{T} \\
& 0 \leq f_{m,t} \leq f_{max} \qquad t \in \mathcal{T} \\
& \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[Q_{m,t}\right] \leq \varepsilon
\end{aligned} \tag{8}$$

P1 is a long-term optimization problem that each MU aims to solve. More concretly, MU's aim is to minimize its expected energy consumption (with respect to the stochastic r.v. $A_{m,t}$ and $\{g_{m,k,t}\}$) while guaranteeing a few constraints. Also, we need to remember that, we have introduced a new

[2]Please note that in **P1**, all the subscripts m are fixed since it is a payoff function for a specific MU.

constraint (i.e., the last one), barring those we have already mentioned before. The constraint could be viewed as a long-term "soft" constraint, which is set for the purpose of maintaining acceptable delay for handling the workload. A positive constant ε in the constraint serves as the upper bound of the long-term average queue backlog, which is set according to the preference of the MUs.[3] Then, in view of Little's law (see [12]), if the long-term expected backlog of the workload queue is being bounded by a constant, we can safely conclude a finite waiting time in the queue (or latency) and the latency is positively correlated with the upper bound value of ε. Maintaining a bounded latency for workload should be a basic requirement for the MU and this explains why we need the last constraint. But it is worth noting that such a problem is impossible to solve offline. The obstacle lies in our lacking knowledge on the stochastic workload arrival $A_{m,t}$ and the channel gain $\{g_{m,k,t}\}$, which makes it impossible to measure the expected queue length and energy consumption in advance. In this regard, we shall have a light transformation of the long-term payoff function for ease of further solving.

3.5 Online Payoff and Its Solution Space

Now we shall transform the long-term payoff to a slot-by-slot online one for a substitution. Note that, the lately arrived workload A_t, the current queue length $Q_{m,t}$ and the channel condition $\{g_{m,k,t}\}$ are all revealed to MUs at the beginning of a time slot. In order to maintain the stability of the workload queues, we could minimize the difference of square for the queues (more precisely, its upper-bound) by carefully adjusting the consumed workload. Formally, the difference of square and its upper-bound can be given by:

$$
\begin{aligned}
\Delta(Q_{m,t}, A_{m,t}, \{g_{m,k,t}\}) &= \frac{1}{2}(Q_{m,t+1}^2 - Q_{m,t}^2) \\
&\leq \frac{1}{2}\left((Q_{m,t} + A_{m,t} - D_{m,t})^2 - Q_{m,t}^2\right) \\
&= \frac{1}{2}(A_{m,t} - D_{m,t})^2 + Q_{m,t}(A_{m,t} - D_{m,t}) \\
&\leq \frac{1}{2}(A_{max}^2 + D_{max}^2) + Q_{m,t}(A_{m,t} - D_{m,t})
\end{aligned}
\tag{9}
$$

Note that we have added a constant $1/2$ to the standard difference of square, but it would not affect the desirable property of the formulation. Typically, the formulation of $\Delta(Q_{m,t}, A_{m,t}, \{g_{m,k,t}\})$ is called *Lyapunov function* and the upper-bound of which is called *Lyapunov drift* (see [17]). With some detail being omitted (most of which could be found in [17]), we need to clarify that minimizing Lyapunov drift could help maintain the queue stability. This motivates us to transform the long-term "s oft" constraint shown in **P1** (the last one) to the online scheduling objective. Formally, we propose our transferred problem, stating:

$$
\begin{aligned}
\textbf{(P2)} \qquad & \underset{\{p_{m,k,t}\}, f_{m,t}}{minimize} \quad -Q_{m,t}D_{m,t} + V\left(\hat{E}_{m,t} + \sum_{k=1}^{K} \tilde{E}_{m,k,t}\right) \\
s.t. \qquad & \sum_{k=1}^{K} p_{m,k,t} \leq p_{max} \\
& p_{m,k,t} \geq 0 \qquad \forall k \in \mathcal{K} \\
& 0 \leq f_{m,t} \leq f_{max}
\end{aligned}
\tag{10}
$$

[3]For convenience, we assume that all the MUs share the same sensitivity of latency, implying that the queue upper bound constraint ε is uniform for all of them.

where V serves as a tradeoff parameter between minimizing energy consumption and maintaining queue stability. Also, we need to note that the exact setting of V is associated with ε in **P1**. It is also notable that the first term in **P2** is not the direct substitution using $\Delta(Q_{m,t}, A_{m,t}, \{g_{m,k,t}\})$ since we have eliminated the constant term (i.e. $\frac{1}{2}(A_{max}^2 + D_{max}^2)$, $Q_{m,t}A_{m,t}$), but it should not bring any critical impacts towards the problem transformation, except that the setting of tradeoff parameter V could be affected. Additionally, we see that the new payoff majorly differs from **P1** in that we have discomposed a long-term objective into phases objectives for each slot, which shows the essence of our transformation from an offline payoff to an online one. But we also see that the online payoff contains interference brought by other MUs (see our definition of $D_{m,t}$ and Equation (2)), which makes the online scheduling problem still impossible to be solved if others' decisions are intangible.[4] Later, we would introduce our game theory approach to handle the interference issue, but for now, we first need to admit that these particular interferences are known to us. More explicitly, we now make it clear that our next task is to derive an optimal solution for **P2**[5] under the assumption that $p_{m',k,t}$ and $f_{m',t}$ are known for all $m' \neq m$ and the reason we need to do this particular job would be well explained later. Then, to do the optimization, we first split the scheduling problem[6] based upon the two optimization variables, as follows:

$$\begin{aligned} \textbf{(P2-p)} \qquad & \underset{\{p_{m,k,t}\}}{minimize} \quad \sum_{k=1}^{K}\left(-B\log_2\left(1+\frac{p_{m,k,t}g_{m,k,t}}{\sigma^2+\sum_{\substack{m'=1\\m'\neq m}}^{M}p_{m',k,t}g_{m',k,t}}\right)Q_{m,t}\delta + Vp_{m,k,t}\delta\right) \\ s.t. \qquad & \sum_{k=1}^{K}p_{m,k,t} \leq p_{max} \\ & p_{m,k,t} \geq 0 \qquad \forall k \in \mathcal{K} \end{aligned} \tag{11}$$

and

$$\begin{aligned} \textbf{(P2-f)} \qquad & \underset{f_{m,t}}{minimize} \quad -\frac{f_{m,t}\cdot\delta}{C}Q_{m,t} + V\gamma_m f_{m,t}^3\delta \\ s.t. \qquad & 0 \leq f_{m,t} \leq f_{max} \end{aligned} \tag{12}$$

From the problem structure, we can derive the following observation:

Observation 1. *__P2__ reaches its optimality if and only if both __P2-p__ and __P2-f__ reach their optimalities.*

Observation 2. *Solving __P2-f__ does not require knowledge on other MUs' decision while solution of __P2-p__ does require those particular information owing to the existence of $p_{m',k,t}$ in its objective function.*

As per Observation 1, we know that in order to derive the optimality of P2, we shall get access to the exact optimality of both **P2-p** and **P2-f**. Now we first start with the solution of **P2-p**, which is based on the well-known *water-filling* solution.

[4]If the scheduling decisions of other MUs are unknown, then the interference brought by others could not be precisely calculated, making it impossible to derive the data rate, which is presented in the objective function of **P2**.
[5]For this particular online payoff, both the subscripts m and t are fixed, since the online payoff is designated for a specific MU m and a specific time slot t.
[6]Henceforth, we use the term "p roblem" or "s cheduling problem" to call the MU's payoff and its solution space.

*Claim 1 (Optimality for **P2-p**).* If $p_{m,k,t}$ is the optimal solution for **P2-p**, then:

$$\begin{aligned} p_{m,k,t} &= \left[\frac{BQ_{m,t}\delta}{\ln 2(V\delta + v)} - \frac{1}{\alpha(p_{-m,k,t})}\right]^+ \\ \textbf{subject to:} \quad & \sum_{k=1}^{K} p_{m,k,t} - p_{max} \leq 0 \\ & v \cdot \left(\sum_{k=1}^{K} p_{m,k,t} - p_{max}\right) = 0 \\ & v \geq 0. \end{aligned} \tag{13}$$

where $\alpha(p_{-m,k,t}) = \frac{g_{m,k,t}}{\sigma^2 + \sum_{\substack{m'=1 \\ m' \neq m}}^{M} g_{m',k,t} p_{m',k,t}}$.

Proof. Let $\alpha(p_{-m,k,t}) = \frac{g_{m,k,t}}{\sigma^2 + \sum_{\substack{m'=1 \\ m' \neq m}}^{M} g_{m',k,t} p_{m',k,t}}$. Then Karush-Kuhn-Tucker (KKT) conditions for **P2-p** give:

$$\begin{aligned} -\frac{BQ_{m,t}\delta\alpha(p_{-m,k,t})}{\ln 2 \cdot (1 + \alpha(p_{-m,k,t}))p_{m,k,t}} + V\delta - u_k + v = 0 \quad & \forall k \in \mathcal{K} \\ u_k \cdot p_{m,k,t} = 0, \quad u_k \geq 0 \quad & \forall k \in \mathcal{K} \\ v \cdot \left(\sum_{k=1}^{K} p_{m,k,t} - p_{max}\right) = 0, \quad v \geq 0 & \\ p_{m,k,t} > 0, \quad \sum_{k=1}^{K} p_{m,k,t} - p_{max} \leq 0 \quad & \forall k \in \mathcal{K} \end{aligned} \tag{14}$$

where u_k and v are the Lagrange dual variables. By eliminating u_k, it yields:

$$\begin{aligned} \frac{BQ_{m,t}\delta\alpha(p_{-m,k,t})}{\ln 2 \cdot (1 + \alpha(p_{-m,k,t})p_{m,k,t})} - V\delta \leq v & \\ \left(-\frac{BQ_{m,t}\delta\alpha(p_{-m,k,t})}{\ln 2 \cdot (1 + \alpha(p_{-m,k,t})p_{m,k,t})} + V\delta + v\right) \cdot p_{m,k,t} = 0, \quad u_k \geq 0 \quad & \forall k \in \mathcal{K} \\ v \cdot \left(\sum_{k=1}^{K} p_{m,k,t} - p_{max}\right) = 0, \quad v \geq 0 & \\ p_{m,k,t} > 0, \quad \sum_{k=1}^{K} p_{m,k,t} - p_{max} \leq 0 \quad & \forall k \in \mathcal{K} \end{aligned} \tag{15}$$

Then we argue that:

$$p_{m,k,t} = \begin{cases} 0 & v > \left(\frac{BQ_{m,t}\alpha(p_{-m,k,t})}{\ln 2} - V\right)\delta \\ \frac{BQ_{m,t}\delta}{\ln 2(V\delta + v)} - \frac{1}{\alpha(p_{-m,k,t})} & v \leq \left(\frac{BQ_{m,t}\alpha(p_{-m,k,t})}{\ln 2} - V\right)\delta \end{cases} \tag{16}$$

It can also be rewritten as:

$$p_{m,k,t} = \left[\frac{BQ_{m,t}\delta}{\ln 2(V\delta + v)} - \frac{1}{\alpha(p_{-m,k,t})}\right]^+ \tag{17}$$

subject to: $\sum_{k=1}^{K} p_{m,k,t} - p_{max} \leq 0$, $v \cdot (\sum_{k=1}^{K} p_{m,k,t} - p_{max}) = 0$ and $v \geq 0$. Then, we claim that the above condition is the necessary condition for optimality of **P2-p**. □

Claim 1 does not immediately promise us a close-form solution or a traditional water-filling solution (notice that v could be 0). In this regard, we shall discuss two separate cases (i.e., $v = 0$ and $v \neq 0$), in order to transfer it to the solvable form.

Case 1 ($v = 0$). In this case, $p_{m,k,t} = [\frac{BQ_{m,t}}{\ln 2V} - \frac{1}{\alpha(p_{-m,k,t})}]^+$ and we need to check if it meets $\sum_{k=1}^{K} p_{m,k,t} - p_{max} \leq 0$. If the above constraint does not break, we count $p_{m,k,t} = [\frac{BQ_{m,t}}{\ln 2V} - \frac{1}{\alpha(p_{-m,k,t})}]^+$ as one of the candidates of the optimal solution.

Case 2 ($v > 0$). In this case, $p_{m,k,t} = [\frac{BQ_{m,t}\delta}{\ln 2(V\delta+v)} - \frac{1}{\alpha(p_{-m,k,t})}]^+$ and we need to make sure $\sum_{k=1}^{K} p_{m,k,t} - p_{max} = 0$. Then it reduces to a normal water-filling solution and could be solved by bisection method.

Recall that Claim 1 tells us that, if a solution is optimal, then it must be one of the feasible solutions derived from the above two cases. But we must argue that Case 2 could not yield any feasible solution if there exists a feasible solution in Case 1. More concretely, we notice that, for $\sum_{k=1}^{K}[\frac{BQ_{m,t}\delta}{\ln 2(V\delta+v)} - \frac{1}{\alpha(p_{-m,k,t})}]^+$, v is monotonically decreasing over $[0, (\frac{BQ_{m,t}\max_{k\in\mathcal{K}}\{\alpha(p_{-m,k,t})\}}{\ln 2} - V)\delta]$ (if the right end > 0) and non-increasing over $[0, \infty)$. If we have a feasible solution in Case 1, it follows that $\sum_{k=1}^{K}[\frac{BQ_{m,t}\delta}{\ln 2V\delta} - \frac{1}{\alpha(p_{-m,k,t})}]^+ - p_{max} < 0$ OR $\sum_{k=1}^{K}[\frac{BQ_{m,t}\delta}{\ln 2V\delta} - \frac{1}{\alpha(p_{-m,k,t})}]^+ - p_{max} = 0$. For the < 0 case (the previous one), based on the non-decreasing property of v over $[0, \infty)$, it suffices to show that $\sum_{k=1}^{K}[\frac{BQ_{m,t}\delta}{\ln 2(V\delta+v)} - \frac{1}{\alpha(p_{-m,k,t})}]^+ - p_{max} < 0$ for any $v > 0$. For the $= 0$ case (the latter one), we know that $(\frac{BQ_{m,t}\max_{k\in\mathcal{K}}\{\alpha(p_{-m,k,t})\}}{\ln 2} - V)\delta > 0$ and combining the monotonically decreasing property of v over the interval $[0, (\frac{BQ_{m,t}\max_{k\in\mathcal{K}}\{\alpha(p_{-m,k,t})\}}{\ln 2} - V)\delta]$, we get the same result as the previous case. The result indicates that $\sum_{k=1}^{K} p_{m,k,t} - p_{max} = 0$ could not happen for any $v > 0$ if there exist a feasible solution in Case 1. Now we see that the optimal solution could only be derived from either Case 1 or Case 2 (but not together). As such, our algorithm will first check the first case to see that if the constraint is violated. If and only if it does, the algorithm continues to solve the water-filling solution with the bisection method for Case 2. Formally, we show in Algorithm 1 the detailed process and the algorithm have a computation complexity of only $O(\log_2 n)$. Now, we shall show that **P2-f** is also solvable and this time we can derive a close-form solution for it.

*Claim 2 (Optimality for **P2-f**).* **P2-f** reaches its optimality if and only if:

$$f_{m,t} = \min\left\{\sqrt{\frac{Q_{m,t}}{3V\gamma_m C}}, f_{max}\right\} \tag{18}$$

Proof. Taking the first derivative of $\hat{F}(f_{m,t}) = -\frac{f_{m,t}\cdot\delta}{C}Q_{m,t} + V\gamma_m\delta f_{m,t}^3$, it yields:

$$\hat{F}'(f_{m,t}) = -\frac{\delta}{C}Q_{m,t} + 3V\gamma_m\delta f_{m,t}^2 \tag{19}$$

Making $\hat{F}'(f_{m,t}) = 0$, we have $f_{m,t} = \sqrt{\frac{Q_{m,t}}{3V\gamma_m C}}$. Then we know that $\hat{F}(f_{m,t})$ is strictly decreasing in $[0, \sqrt{\frac{Q_{m,t}}{3V\gamma_m C}}]$ and increasing in $[\frac{Q_{m,t}}{3V\gamma_m C}, \infty)$. Making the constraint $f_{m,t} \leq f_{max}$ effective, it brings us the final results. □

In a nutshell, based on Claim 1 and 2, we can derive the following observation for the optimal solutions of **P2**:

ALGORITHM 1: Optimal solution for **P2-p**

Input:
Length of per time slot; δ
Power coefficient; $\{\alpha(p_{-m,k,t})\}_{k\in\mathcal{K}}$
Convergence criteria for bisection; ϵ_{bi}
Workload queue's backlog; $Q_{m,t}$
Tradeoff parameter; V
Output:
The solution for **P2-p** in round t; $\{p_{m,k,t}\}_{k\in\mathcal{K}}$

if $\sum_{k=1}^{K}\left[\frac{BQ_{m,t}}{\ln 2V}-\frac{1}{\alpha(p_{-m,k,t})}\right]^{+}-p_{max}\le 0$ **then**
  $\{p_{m,k,t}\}_{k\in\mathcal{K}}\leftarrow\left\{\left[\frac{BQ_{m,t}}{\ln 2V}-\frac{1}{\alpha(p_{-m,k,t})}\right]^{+}\right\}_{k\in\mathcal{K}}$
  STOP
end if
$v_r\leftarrow\left(\frac{BQ_{m,t}\max_{k\in\mathcal{K}}\{\alpha(p_{-m,k,t})\}}{\ln 2}-V\right)\delta,\ v_l\leftarrow 0$
while True **do**
  $v_c\leftarrow(v_l+v_r)/2$
  if $\sum_{k=1}^{K}\left[\frac{BQ_{m,t}\delta}{\ln 2(V\delta+v_c)}-\frac{1}{\alpha(p_{-m,k,t})}\right]^{+}-p_{max}=0$ OR $v_r-v_l\le\epsilon_{bi}$ **then**
    $\{p_{m,k,t}\}_{k\in\mathcal{K}}\leftarrow\left\{\left[\frac{BQ_{m,t}\delta}{\ln 2(V\delta+v_c)}-\frac{1}{\alpha(p_{-m,k,t})}\right]^{+}\right\}_{k\in\mathcal{K}}$
    STOP
  end if
  if $sign\left(\sum_{k=1}^{K}\left[\frac{BQ_{m,t}\delta}{\ln 2(V\delta+v_c)}-\frac{1}{\alpha(p_{-m,k,t})}\right]^{+}-p_{max}\right)=sign\left(\sum_{k=1}^{K}\left[\frac{BQ_{m,t}\delta}{\ln 2(V\delta+v_l)}-\frac{1}{\alpha(p_{-m,k,t})}\right]^{+}-p_{max}\right)$ **then**
    $v_l\leftarrow v_c$
  else
    $v_r\leftarrow v_c$
  end if
end while

Observation 3. *First, with a bigger setting of V, the optimal solution of* ***P2*** *tends to contain a smaller $f_{m,t}$ and $p_{m,k,t}$. Second, with a bigger $\alpha(p_{-m,k,t})$ for a UAV k, we are more likely to have a larger $p_{m,k,t}$ over the designated UAV though it has no impact on $f_{m,t}$. Finally, a larger $Q_{m,t}$ brings us a bigger $f_{m,t}$ and $p_{m,k,t}$.*

The above observation is important and it is consilient with our expectations. First, with a bigger tradeoff parameter V, the MU tends to save energy, so it should lower the CPU frequency and transmission power to reach this goal, which is to say, $f_{m,t}$ and $p_{m,k,t}$ should be lowered. Second, with a bigger power coefficient of UAV k, which indicates that its current channel condition is good, MUs tend to escalate the transmission power to this particular UAV. Finally, if the queue backlog $Q_{m,t}$ is big, the MU needs to raise the transmission power and the CPU frequency so that the backlog could be well contained. Therefore, as we mentioned in Section 3.3, it is likely that $D_{m,t}>Q_{m,t-1}+A_{m,t}$ would happen if V is set to an extremely low value, so that the dummy workload would be sent for computation or offloading. But if V is set to an accommodating value, we see that the CPU frequency and transmission power should not be much higher, and so that the amount of dummy workload being scheduled could be well contained.[7]

The two proposed sub-problems are all convex and we see that the solving of **P2-f** does not rely on knowledge on other MU's decisions while the solving of **P2-p** does. In other words, other MUs' decision might directly affect the radio power allocation of an MU, resulting in a decision deadlock between MUs. To cope with the issue, in the next section, we shall show a game-theory-based approach that allows us to reach a Nash Equilibrium (NE) via an iterative solving of **P2**.

[7]In fact, we need to claim here that, in most cases, the undesirable redundant scheduling should not happen.

3.6 Extraction of Nash Equilibrium

In the previous section, we have formulated the online payoff of MUs and we also succeed to derive a reliable solution for an MU under the condition that other's decisions are known. With the work being done, we are able to derive a *Nash Equilibrium* (NE) under a non-cooperative game setting, where every player only cares about his or her own interest, namely, each individual strives to minimize their independent payoff function. Note that in a non-cooperative game, we assume that the MUs would neither compromise their own interest nor group together to reach a social benefit. Now we formally introduce the definition of *Nash Equilibrium* for each round under the non-cooperative game setting, as follows:

Definition 1 (Nash Equilibrium). A strategy profile $\boldsymbol{s}_t^* \triangleq \{\{p_{m,k,t}\}_{k\in\mathcal{K}}, f_{m,t}\}_{m\in\mathcal{M}}$ is called a Nash Equilibrium for round t when no player in the game could proactively reduce its own payoff by changing its strategy $\boldsymbol{s}_{m,t} \triangleq (\{p_{m,k,t}\}_{k\in\mathcal{K}}, f_{m,t})$ while other players' strategies $\boldsymbol{s}_{-m,t} \triangleq \{\{p_{m',k,t}\}_{k\in\mathcal{K}}, f_{m',t}\}_{m'\in\mathcal{M}-m}$ are fixed, namely, we have:

$$L(\boldsymbol{s}_{m,t}^*, \boldsymbol{s}_{-m,t}^*) \leq L(\boldsymbol{s}_{m,t}, \boldsymbol{s}_{-m,t}^*) \quad \forall \boldsymbol{s}_m \in \mathcal{S}_m \tag{20}$$

where $L(\boldsymbol{s}_{m,t}, \boldsymbol{s}_{-m,t}) \triangleq -Q_{m,t}D_{m,t} + V(\hat{E}_{m,t} + \sum_{k=1}^{K} \tilde{E}_{m,k,t})$ is the online payoff in **P2**. $\mathcal{S}_m$ is the solution space of MU m, which is qualified by the constraints in **P2**. Informally, we can regard an NE as a coordinated strategy for each selfish player in the game, where all of the players would find such a scheduling result acceptable since they simply could not benefit more by changing its strategy. From the perspective of a service operator (or the UAV's operator) whose aim is to provide consistent services and resource regulation for all the MUs, it is quite attractive for him or her to extract such an equilibrium and publicize it as a regulation decision for the coming round, such that each MU would have no reason to renege and would certainly follow the regulation. Also, an NE is known to be a socially beneficial scheme that is capable of reducing resource competition, which makes it even more tempting for the service provider. Now, we are sufficiently motivated to find an approach to derive an NE for the service provider, but we first need to make sure the existence of at least one of which for the game. To show this, we state:

THEOREM 1 (EXISTENCE OF NASH EQUILIBRIUM). *There exists at least one Nash Equilibrium for round t if and only if the payoff function, $L(\boldsymbol{s}_{m,t}, \boldsymbol{s}_{-m,t})$, is convex of $\boldsymbol{s}_{m,t}$ with fixed $\boldsymbol{s}_{-m,t}$ and the solution space $\mathcal{S}_m$ is convex, closed, and bounded.*

Theorem 1 is a classical result from [19] and it has been widely used to prove the existence of the Nash Equilibrium (e.g., [10]). Revisiting **P2**, we are confident to say that all conditions listed in Theorem 1 holds for our case, and thus, we claim that there is at least one well-defined Nash Equilibrium in our game.

Now we shall formally introduce our way to extract one of the equilibriums for a round t, as presented in Algorithm 2. In Algorithm 2, $\left\|\boldsymbol{p}_t^q - \boldsymbol{p}_t^{q-1}\right\| = \sqrt{\sum_{m=1}^{M}\sum_{k=1}^{K}\left(p_{m,k,t}^q - p_{m,k,t}^{q-1}\right)^2}$ measures if the algorithm has reached convergence. One might notice that we adopt a rather simple way to achieve the *Nash Equilibrium*. We simply simulate a bargaining process for the MUs, where we minimize the payoff of different MUs in a sequence and repeat the process until all the MUs would rather stay with their old choices in the last iteration. Such a bargaining process is known to be an efficacious way to derive a Nash Equilibrium, as long as it does exist. Besides, it is noticeable that we only involve the variables regarding transmission power allocation (i.e., $\{p_{m,k,t}\}$) into the bargaining process, but the other groups of variables (i.e., $f_{m,t}$) do not participate the process. The idea behind could be derived from our previous statement in Observation 2, which implies that MU's decisions of CPU frequency would not be affected by others.

ALGORITHM 2: Gauss-Seidel-type method for Nash Equilibrium

Input:

Length of per time slot; δ
Convergence criteria for extraction; ϵ_{ne}
Channel gain ; $\{g_{m,k,t}\}_{m\in\mathcal{M},k\in\mathcal{K}}$
Queues' backlogs; $\{Q_{m,t}\}_{m\in\mathcal{M}}$
Tradeoff parameter; V

Output:

Nash equilibrium for round t ; $\{\boldsymbol{s}_t^*\}$

Initialize iteration $q = 0$
for $m \in \mathcal{M}$ **do**
  Initialize $\boldsymbol{p}_{m,t}^q \leftarrow (0, 0, \dots, 0)_K$, $f_{m,t}^q \leftarrow 0$
end for
repeat
  $q = q + 1$
  for $m \in \mathcal{M}$ **do**
    Calculate $\{\alpha(p_{-m,k,t})\}_{k\in\mathcal{K}}$ based upon $\{g_{m,k,t}\}_{m\in\mathcal{M},k\in\mathcal{K}}$ and $\left(\boldsymbol{p}_{1,t}^q, \dots, \boldsymbol{p}_{m-1,t}^q, \boldsymbol{p}_{m+1,t}^{q-1}, \dots, \boldsymbol{p}_{M,t}^{q-1}\right)$
    $\boldsymbol{p}_{m,t}^q \leftarrow$ execute Algorithm 1 with $\{\alpha(p_{-m,k,t})\}_{k\in\mathcal{K}}, Q_{m,t}$.
  end for
  $\boldsymbol{p}_t^q \leftarrow \left\{\boldsymbol{p}_{m,t}^q\right\}_{m\in\mathcal{M}}$
until $\left\|\boldsymbol{p}_t^q - \boldsymbol{p}_t^{q-1}\right\| \le \epsilon_{ne}$
for $m \in \mathcal{M}$ **do**
  $f_{m,t} \leftarrow \min\left\{\sqrt{\frac{Q_{m,t}}{3V\gamma_m C}}, f_{max}\right\}$
end for
$\boldsymbol{s}_t^* \leftarrow \left\{\boldsymbol{p}_{m,t}^q, f_{m,t}\right\}_{m\in\mathcal{M}}$

With a successful extraction of the Nash Equilibrium for each round, now we shall show the overall online scheduling procedure for the service (or UAV) operators, as shown in Algorithm 3. The scheduling strategy called the Online Non-Cooperative Computation Offloading (ONCCO) could be run by a master UAV or a centralized MEC server, and the decision should be enforced by the MUs thanks to the basic rationale behind the Nash Equilibrium.

4 STAGE TWO: WORKLOAD SCHEDULING FOR UAVS

4.1 Problem Formulation of Stage Two

Now we have derived a scheduling solution for the MUs and based on the scheduling decision, we gain a chance to dynamically adjust the UAV in order accomodate the incoming offloading workload. Now, we will have a discussion on how to schedule the computation capability of UAVs given the Nash Equilibrium obtained in Stage One. First, we show that the amount of arrival workload of a UAV k in round t can be formulated as follows:

$$\lambda_{k,t} = \sum_{m\in\mathcal{M}} B\log_2\left(1 + \frac{p_{m,k,t}g_{m,k,t}}{\sigma^2 + \sum_{\substack{m'=1\\ m'\neq m}}^{M} p_{m,k,t}g_{m,k,t}}\right)\delta \tag{21}$$

ALGORITHM 3: Online Non-Cooperative Computation Offloading (ONCCO)

Input:
Length of per time slot; δ
Output:
Scheduling Decision; $\{\boldsymbol{s}_t\}_{t\in\mathcal{T}}$

1: Initialize $Q_{m,t} \leftarrow 0$ for all $m \in \mathcal{M}$
2: **for** $t \in \{1, 2, \ldots, \}$ **do**
3: MUs proactively report arrival workload $A_{m,t}$
4: Calculate $\{g_{m,k,t}\}_{m\in\mathcal{M},k\in\mathcal{K}}$ based upon the observed geographical information (see Equation (1))
5: $\boldsymbol{s}_t \leftarrow$ execute Algorithm 2 with $\{g_{m,k,t}\}_{m\in\mathcal{M},k\in\mathcal{K}}$ and $\{Q_{m,t}\}_{m\in\mathcal{M}}$
6: Disclose $\boldsymbol{s}_t$ to all the MUs.
7: UAVs receive the scheduled workloads and do the computation
8: Update $Q_{m,t+1}$ according to Equation (7) for all MU $m \in \mathcal{M}$ using $\boldsymbol{s}_t$ and A_t
9: **end for**

Here, we argue that $\lambda_{k,t}$ can be precisely calculated given the achieved Nash Equilibrium, and in this standard, we simply regard it as a known constant for us.

In our UAV scheduling setting, we assume that a UAV is equipped with multiple CPU cores, captured by a set $\mathcal{N}_k = \{1, 2, \ldots, N_k\}$ and its index is denoted by n. In addition, we assume the **Dynamic Voltage and Frequency Scaling (DVFS)** technique is adopted in UAVs, so the CPU frequency of each core can be dynamically adjusted in order to achieve energy conservation. Like Equation (4), we can derive the amount of data being served during a time slot t, as follows:

$$\bar{D}_{k,t} = \sum_{n=1}^{N_k} \frac{\bar{f}_{k,n,t} \cdot \delta}{C} \tag{22}$$

where C denotes the required CPU cycles for computation of per unit of data and $\bar{f}_{k,n,t}$ is the CPU frequency of core n of UAV k.

The corresponding energy consumption can be formally formulated as:

$$\bar{E}_{k,t} = \sum_{n=1}^{N_k} \bar{\gamma}_k \bar{f}^3_{k,n,t} \delta, \tag{23}$$

where $\bar{\gamma}_k$ denotes the effective capacitance coefficient of the UAV k. Also, we have a maximum cap for $\bar{f}_{k,n,t}$, i.e., $0 < \bar{f}_{k,n,t} < \bar{f}_{max}$.[8]

Like the workload queues of MUs, we also maintain a queue for each UAV, whose backlog evolves according to:

$$\bar{Q}_{k,t+1} = \max\left\{\bar{Q}_{k,t} - \bar{D}_{k,t}, 0\right\} + \lambda_{k,t} \tag{24}$$

[8]For simplicity, we assume each UAV is equipped with homogeneous CPU cores, which implies that their effective capacitance coefficient and maximum frequency are the same. But we note that our model could be easily extended to the heterogeneous setting with only a minor overhaul.

Then we shall give a long-term UAV computation resources scheduling problem for a specific UAV, which is quite similar to **P1**, as follows:

$$\begin{aligned} &\textbf{(P3)} \quad \underset{\{f_{k,n,t}\}}{minimize} \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[\bar{E}_{k,t}\right] \\ &s.t. \quad 0 \le \bar{f}_{k,n,t} \le \bar{f}_{max} \qquad n \in \mathcal{N}_k, t \in \mathcal{T} \\ &\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[\bar{Q}_{k,t}\right] \le \bar{\varepsilon} \end{aligned} \tag{25}$$

where $\bar{\varepsilon}$ is a positive constant that maintains the time-average queue backlogs. Similarly, we shall transform the problem into an online one, following the same technique with the transformation of **P1**. Formally, we have:

$$\begin{aligned} &\textbf{(P4)} \quad \underset{\{f_{k,n,t}\}}{minimize} \sum_{n=1}^{N_k} -\bar{Q}_{k,t} \frac{\bar{f}_{k,n,t}\delta}{C} + \bar{V}\bar{\gamma}_k \bar{f}_{k,n,t}^3 \delta \\ &s.t. \qquad 0 \le \bar{f}_{k,n,t} \le \bar{f}_{max} \qquad n \in \mathcal{N}_k \end{aligned} \tag{26}$$

where $\bar{V}$ is the tradeoff parameter between UAV's energy consumption and its queue stability. Then it is not hard to derive a close-form solution for this problem, as given in Claim 3.

*Claim 3 (Optimality for **P4**).* **P4** reaches its optimality if and only if:

$$\bar{f}_{k,n,t} = \min\left\{\sqrt{\frac{\bar{Q}_{k,t}}{3\bar{V}\bar{\gamma}_k C}}, \bar{f}_{max}\right\} \tag{27}$$

The proof of this claim is basically the same with Claim 2 and thus is omitted here.

4.2 Algorithm

With Claim 3, we could immediately show our algorithm for UAV workload scheduling (called WS-UAV), in Algorithm 4.

5 NUMERICAL SIMULATION AND RESULTS

5.1 Simulation Setting

Our proposed algorithms are evaluated in a simulation environment. Now we first introduce the setting of the environment. In our simulation, the position of each UAV and MU is confined in a restricted area. Formally, we have the following constraint on their two dimensional coordinate, $x_{m,t}, y_{m,t} \in [0, 100]$ and $x_{k,t}, y_{k,t} \in [0, 100]$. To simulate the mobility of both UAVs and MUs, for every time slot t, we have $x_{m,t+1} = [x_{m,t} + \Delta^x_{m,t}]_0^{100}$, $y_{m,t+1} = [y_{m,t} + \Delta^y_{m,t}]_0^{100}$, $[x_{k,t+1} = x_{k,t} + \Delta^x_{k,t}]_0^{100}$ and $[y_{k,t+1} = y_{k,t} + \Delta^y_{k,t}]_0^{100}$ where $[\cdot]_0^{100}$ confines the coordinate within $[0, 100]$m and $\Delta^x_{m,t}$, $\Delta^y_{m,t}$, $\Delta^x_{k,t}$, $\Delta^y_{k,t}$ independently sample from a uniform distribution within $[-10, 10]m$. The initial position of MUs and UAVs are set randomly within the restricted area. To model the workload arrival of each time slot, we assume $A_{m,t} = \eta \cdot 10^6$ bits where η is an independent r.v. following an exponential distribution with its expectation = 4. Other setting of system parameters can be found in Table 2.

5.2 Performance Evaluation on ONCCO

To conduct a comprehensive evaluation of offloading on stage 1, we first prepare a few baseline schemes for comparison, as follows:

ALGORITHM 4: Workload Scheduling for UAV (WS-UAV)

Input:
Tradeoff parameter for UAV workload scheduling; $\bar{V}$
Output:
Scheduled CPU frequency; $\{\bar{f}_{k,n,t}\}$

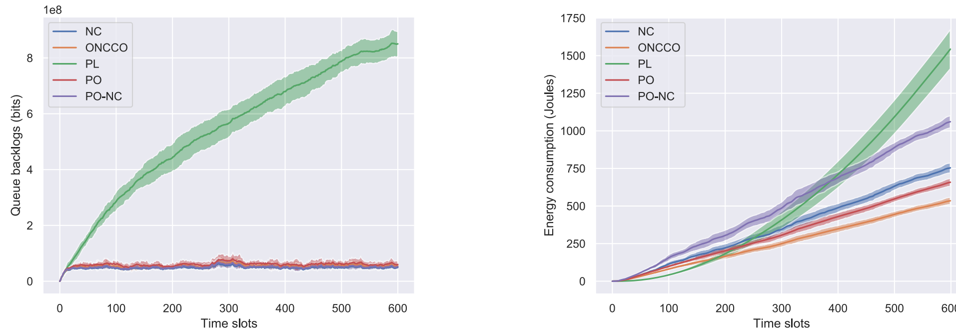
1: Initialize $\bar{Q}_{k,t} \leftarrow 0$ for $k \in \mathcal{K}$
2: **for** $t \in \{1, 2, \ldots, \}$ **do**
3:   **for** UAV $k \in \mathcal{K}$ **in parallel do**
4:     Calculate the arrival workload $\lambda_{k,t}$ based on the Nash Equilibrium obtained in Algorithm 3
5:     **for** $n \in \mathcal{N}_k$ **do**
6:       $\bar{f}_{k,n,t} \leftarrow \min\left\{\sqrt{\frac{\bar{Q}_{k,t}}{3\bar{V}\bar{\gamma}_k C}}, \bar{f}_{max}\right\}$
7:     **end for**
8:     Update $\bar{Q}_{k,t+1}$ according to Equation (24)
9:   **end for**
10: **end for**

Table 2. Experimental Setting

Notations	Values	Notations	Values
M	12	N_k	5
K	3	g_0	−50 dB
p_{max}	1 W	γ_m, γ_k	10^{-28}
H	80 m	C	5×10^3 cycles/bit
δ	10 s	$f_{max}, \bar{f}_{max}$	2 GHz, 2.5 GHz
B	2.5 MHz	σ^2	10^{-9} W
ϵ_{bi}	10^{-6}	ϵ_{ne}	10^{-4}

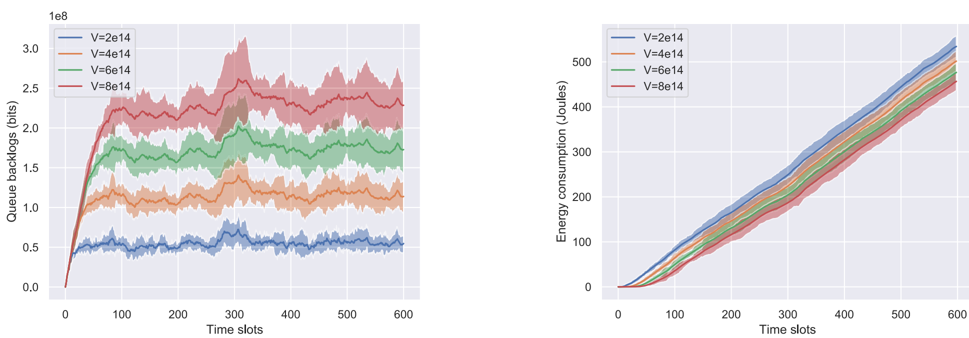
- **N**on-**C**oordinated (NC). In this scheme, MUs do the scheduling themselves in view of their own payoff and because they are not aware of the existence of other MUs, they take the power coefficient $\alpha(p_{-m,k,t}) = g_{m,k}/\sigma^2$. Then, the solution yields by this scheme can be computed based on Claim 1 and Claim 2.
- **P**ure **L**ocal (PL). In this scheme, MUs do not offload workload to UAVs and all the computing is being done by the local CPU. The solution is yielded by Claim 2.
- **P**ure **O**ffloading (PO). This scheme is basically the same with our proposed ONCCO, except that the local CPU frequency is set to 0, indicating that there is no local computing.
- **P**ure **O**ffloading with **N**o **C**oordination (PO-NC). This scheme is mostly the same as the first baseline NC, except that there is no local computing. (i.e., local CPU frequency equals to 0).

5.2.1 Queue Stability and Cumulative Energy Consumption. We then show the variation of queues' backlogs and cumulative energy consumption of different strategies under the same tradeoff parameter $V = 2e14$, as shown in Figure 2. In the figures, the translucent area in the figures marks the standard deviation among MUs and the solid curve depicts the mean of the corresponding value among all the MUs. From (a), we see that the queue backlogs of all the schemes seem



(a) Time vs. queues' backlog (b) Time vs. cumulative energy consumption

Fig. 2. Performance of different scheduling schemes.



(a) Time vs. queues' backlog (b) Time vs. cumulative energy consumption

Fig. 3. V's impact on performance.

to converge to a fixed constant value, though it is noticeable that those of PL shows to be much greater than other schemes. This phenomenon reflects our setting of local computation and transmission capability. Typically, computation of 1 bit of data requires more energy consumption than transmitting it with a channel in good condition, and therefore, the setting of tradeoff parameter V shows to be too large for PL.[9] Overall, we see that our online scheduling schemes ensure the stability of workload queues. And combining the basic knowledge of queue theory, we know that the waiting time in the queues, which is proportional to the queue backlogs, is not unbounded. While the other schemes barring PL share a similar convergent value of queue backlog, it is interesting to see that they have disparate behaviors in terms of energy conservation. We found that our proposed scheme ONCCO basically outperforms others and we also show that the two non-coordinated schemes, PO-NC and NC have additional energy loss, compared to their counterparts, PO and ONCCO. The only difference between the two groups is that the non-coordinated group follows a selfish and myopic decision process and they do not undergo the bargaining process for a Nash Equilibrium. This results in a serious wireless resource competition and causes undesirable performance degradation, i.e., spending more energy for transmission of the same amount of data. Based on the results, we are able to corroborate the social benefit of a Nash Equilibrium.

[9]Typically, a greater setting of V would accelerate the flattening of queue backlogs to some extent, but the energy consumption would escalate at the same time.

Table 3. Mean Rounds of Convergence Under Different ϵ_{ne} and ϵ_{bi}

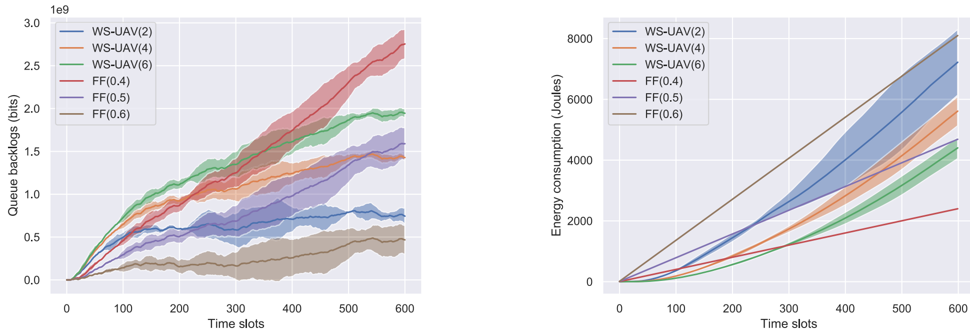
ϵ_{ne} \ ϵ_{bi}	1e-2	1e-3	1e-4	1e-5	1e-6
1e-1	4.17	4.17	4.17	4.17	4.17
1e-2	11.61	11.78	11.76	11.76	11.76
1e-3	inf	25.97	25.98	25.98	25.98
1e-4	inf	inf	38.61	38.6	38.6
1e-5	inf	inf	inf	38.63	38.63

5.2.2 Convergence of Algorithm 2. Then we need to show the convergence of the Gauss-Seidle-type bargaining process in Algorithm 2. The experiment is done by $V = 2e14$ and the simulation steps are 600 in total. We show the data in Table 3 where the figures show the mean rounds of convergence (over the 600 simulation steps) under the convergence criteria ϵ_{ne} and ϵ_{bi}. The inf in the table shows that a deadlock may exist in the process and thus the convergence criteria might not be able to reach. From the data, we see that a smaller ϵ_{ne} necessitates more rounds of bargain, which is quite a common sense, but we also find that ϵ_{bi}, which serves as a convergence criterion for the water-filling process in Algorithm 1, plays a crucial role in the convergence of Algorithm 2. A coarse-grained solution for Algorithm 1 might not be able to promise us a convergence in Algorithm 2. The rationale behind is that Gauss-Seidle-type method is sure to reach convergence if and only if the optimality of all the MUs payoff function is being accessed. Nonetheless, we note that a water-filling solution could only bring us a solution that is very close to the optimality, and therefore its accuracy could affect the process of searching for Nash Equilibrium. Luckily, a bi-section method we used to search for a water-filling solution is known to be of high efficiency and it boasts the computation complexity of only $O(\log_2 n)$, thereby making our proposed scheme applicable in practice. For other scenarios that we could only guarantee a coarse-grain solution, e.g. a heuristic algorithm or an evolutional algorithm, it is hard to say the bargaining process could effectively work.

5.2.3 Impact of V. Another crucial parameter of our proposed ONCCO is the tradeoff parameter V. We show how the algorithm works on different settings of V, as in Figure 3. From (a), we see that all the curves eventually maintain stable, barring some acceptable fluctuation, and the convergence values of queue backlogs are shown to be positively correlated with V. Besides, from (b), we show that the cumulated energy consumption is also positively correlated with V. This reminds us of the purpose of introducing V, which is to control the tradeoff between energy consumption and latency (as reflected by queue backlogs). Also, it is noticeable that our setting of V seems to be quite an enormous figure (with an exponent figure 10^{14}). Our explanation for this is that we keep data measured by bits in the workload queues so the number for the queue backlog is already sufficiently large, prompting that a big number for V is necessitated. Other measurements (e.g. Mbs, Gbs) would also work and its corresponding tradeoff parameter should be lowered. In the last place, although it is hard to be told by the experimental results, we shall claim that, theoretically, the queue backlogs would eventually converge to a bounded value no matter what our setting of V is. A very similar conclusion can be found in [17].

5.3 Performance Evaluation on WS-UAV

Now we commence our experimental evaluation of the workload scheduling on stage two. Firstly, we give a rather naive baseline, termed **F**ixed **F**requency (FF) for comparison purpose. In FF, the



(a) Time vs. queues' backlog (b) Time vs. cumulative energy consumption

Fig. 4. Performance of different UAV workload scheduling schemes.

usage of a UAV's CPU frequency is fixed in advance to a certain ratio and we let β indicate that particular ratio. Formally, for FF(β), we have $\bar{f}_{k,n,t} = \beta f_{max}$ for all k, n, and t. In our experiment on stage 2, we take the same scheduling result produced by ONCCO under $V = 2 \cdot 10^{14}$ as the UAV's workload. Now we show our experimental results in Figure 4 where WS-UAV(x) represents our proposed WS-UAV with the setting of $\bar{V} = x$. It is interesting to note from sub-figure (a) that FF(0.4) and FF(0.5) do not work well, as the stability of their queue backlogs is seriously undermined. For a queue with an unbounded backlog, we know that the computation latency of the workload would become infinity and thereby is unacceptable for any service providers. For FF(0.6), although the algorithm does work in this setting (the queue backlogs seems to be converged and stable), we argue that FF might hardly work in real practice. For one thing, the workload arrival in the real setting is diverse in spatial scales, and therefore, operators might need to undertake laborious testing for the frequency ratio over different spots. For another, the workload arrival might not be stationary and uniform in a temporal sense, for example, the server might face the risk of collapse due to the encounter of a peak hour, though the server itself might be endowed with the full ability to handle the traffic (its CPU frequency is fixed to a normal level in order to save energy). By contrast, our proposed WS-UAV could better handle the hotspot issue. As we have stressed before, our proposed online scheduling scheme could well maintain the queue backlog within a finite value. When a hotspot being encountered, the queue backlogs might experience a temporary rise to a high point, but our algorithm is able to correspondingly adjust the CPU frequency according to the extra traffic and thus making the backlog maintaining at an acceptable level.

6 CONCLUSION AND FUTURE WORKS

In this article, we propose a two-stage optimization scheme for UAV-Assisted computation offloading. In Stage One, we propose a game theory-based solution for service providers, in order to regulate the MU's blind competition over the limited radio resource. With a series of analyses, we confirm that our proposed solution enables service providers to furnish a socially beneficial and individually rational radio power allocation scheme to MUs within the system. In Stage Two optimization, we aim to address the problem of computation resource provision for UAVs, on the basis of the obtained Nash equilibrium. The proposed solution for Stage Two, though exhibits only minor novelty in a technical sense, shows good performance in terms of energy conservation for UAVs.

For our future works, we are thinking about an extension of our system model for Stage Two optimization. One could see that our scheduling of UAV workload is currently running independently for each UAV and we do not allow UAVs to collaborate in order to reach load balance. Factually, it is a realistic assumption on UAVs that they could make use of a designated channel to exchange their workload, in order to balance the load of different UAVs. By adopting the new model, we tend to believe that the UAVs colony, which works as a whole, could achieve better energy conservation performance while ensuring the service standard.

ACKNOWLEDGMENTS

The authors are grateful to the anonymous reviewers for their valuable comments and suggestions.

REFERENCES

[1] Xiaowen Cao, Jie Xu, and Rui Zhang. 2018. Mobile edge computing for cellular-connected UAV: Computation offloading and trajectory optimization. In *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 1–5.

[2] Arcangelo Castiglione, Francesco Palmieri, Ugo Fiore, Aniello Castiglione, and Alfredo De Santis. 2015. Modeling energy-efficient secure communications in multi-mode wireless mobile devices. *J. Comput. System Sci.* 81, 8 (2015), 1464–1478. https://doi.org/10.1016/j.jcss.2014.12.022

[3] A. Castiglione, A. D. Santis, A. Castiglione, F. Palmieri, and U. Fiore. 2013. An energy-aware framework for reliable and secure end-to-end ubiquitous data communications. In *2013 5th International Conference on Intelligent Networking and Collaborative Systems*. 157–165.

[4] Luca Caviglione and Alessio Merlo. 2012. The energy impact of security mechanisms in modern mobile devices. *Network Security* (2012), 2 , 11–14. https://doi.org/10.1016/S1353-4858(12)70015-6

[5] Weiwei Chen, Dong Wang, and Keqin Li. 2018. Multi-user multi-task computation offloading in green mobile edge cloud computing. *IEEE Transactions on Services Computing* (2018).

[6] Xu Chen. 2015. Decentralized computation offloading game for mobile cloud computing. *IEEE Transactions on Parallel and Distributed Systems* 26, 4 (2015), 974–983.

[7] Xu Chen, Lei Jiao, Wenzhong Li, and Xiaoming Fu. 2016. Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Transactions on Networking* 24, 5 (2016), 2795–2808.

[8] Fen Cheng, Shun Zhang, Zan Li, Yunfei Chen, Nan Zhao, F. Richard Yu, and Victor C. M. Leung. 2018. UAV trajectory optimization for data offloading at the edge of multiple cells. *IEEE Transactions on Vehicular Technology* 67, 7 (2018), 6732–6736.

[9] Seongah Jeong, Osvaldo Simeone, and Joonhyuk Kang. 2018. Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning. *IEEE Transactions on Vehicular Technology* 67, 3 (2018), 2049–2063.

[10] Keqin Li. 2018. A game theoretic approach to computation offloading strategy optimization for non-cooperative users in mobile edge computing. *IEEE Transactions on Sustainable Computing* (2018).

[11] R. Li, Z. Zhou, X. Chen, and Q. Ling. 2019. Resource Price-Aware Offloading for Edge-Cloud Collaboration: A Two-Timescale Online Control Approach. *IEEE Transactions on Cloud Computing* (2019), 1–1. https://doi.org/10.1109/TCC.2019.2937928

[12] John D. C. Little and Stephen C. Graves. 2008. Little's law. In *Building Intuition*. Springer, 81–100.

[13] Alessio Merlo, Mauro Migliardi, and Luca Caviglione. 2015. A survey on energy-aware security mechanisms. *Pervasive and Mobile Computing* 24 (2015), 77–90. https://doi.org/10.1016/j.pmcj.2015.05.005 Special Issue on Secure Ubiquitous Computing.

[14] Mohamed-Ayoub Messous, Amel Arfaoui, Ahmed Alioua, and Sidi-Mohammed Senouci. 2017. A sequential game approach for computation-offloading in an UAV network. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 1–7.

[15] Mohamed-Ayoub Messous, Hichem Sedjelmaci, Noureddin Houari, and Sidi-Mohammed Senouci. 2017. Computation offloading game for an UAV network in mobile edge computing. In *2017 IEEE International Conference on Communications (ICC)*. IEEE, 1–6.

[16] Mohamed-Ayoub Messous, Sidi-Mohammed Senouci, Hichem Sedjelmaci, and Soumaya Cherkaoui. 2019. A game theory based efficient computation offloading in an uav network. *IEEE Transactions on Vehicular Technology* 68, 5 (2019), 4964–4974.

[17] Michael J. Neely. 2010. Stochastic network optimization with application to communication and queueing systems. *Synthesis Lectures on Communication Networks* 3, 1 (2010), 1–211.

[18] Lingjun Pu, Xu Chen, Guoqiang Mao, Qinyi Xie, and Jingdong Xu. 2019. Chimera: An Energy-Efficient and Deadline-Aware Hybrid Edge Computing Framework for Vehicular Crowdsensing Applications. *IEEE Internet of Things Journal* 6, 1 (Feb. 2019), 84–99. https://doi.org/10.1109/JIOT.2018.2872436

[19] J. Ben Rosen. 1964. Existence and uniqueness of equilibrium points for concave n-person games. (1964).

[20] Xinhou Wang, Kezhi Wang, Song Wu, Sheng Di, Hai Jin, Kun Yang, and Shumao Ou. 2018. Dynamic resource scheduling in mobile edge cloud with cloud radio access network. *IEEE Transactions on Parallel and Distributed Systems* 29, 11 (2018), 2429–2445.

[21] Yanting Wang, Min Sheng, Xijun Wang, Liang Wang, and Jiandong Li. 2016. Mobile-edge computing: Partial computation offloading using dynamic voltage scaling. *IEEE Transactions on Communications* 64, 10 (2016), 4268–4282.

[22] Qingqing Wu, Yong Zeng, and Rui Zhang. 2018. Joint trajectory and communication design for multi-UAV enabled wireless networks. *IEEE Transactions on Wireless Communications* 17, 3 (2018), 2109–2121.

[23] Hui Xia, Ligang He, Bin Wang, Cheng Chang, Xie Han, and Carsten Maple. 2018. Developing Offloading-enabled Application Development Frameworks for Android Mobile Devices. In *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. IEEE, 416–421.

[24] Changsheng You, Kaibin Huang, Hyukjin Chae, and Byoung-Hoon Kim. 2017. Energy-efficient resource allocation for mobile-edge computation offloading. *IEEE Transactions on Wireless Communications* 16, 3 (2017), 1397–1411.

[25] Jiao Zhang, Li Zhou, Qi Tang, Edith C.-H. Ngai, Xiping Hu, Haitao Zhao, and Jibo Wei. 2018. Stochastic computation offloading and trajectory scheduling for UAV-assisted mobile edge computing. *IEEE Internet of Things Journal* 6, 2 (2018), 3688–3699.

[26] Kaiyuan Zhang, Xiaolin Gui, Dewang Ren, and Defu Li. 2020. Energy-latency tradeoff for computation offloading in UAV-assisted multi-accessedge computing System. *IEEE Internet of Things Journal* (2020).

[27] Fuhui Zhou, Yongpeng Wu, Rose Qingyang Hu, and Yi Qian. 2018. Computation rate maximization in UAV-enabled wireless-powered mobile-edge computing systems. *IEEE Journal on Selected Areas in Communications* 36, 9 (2018), 1927–1941.

[28] Zhi Zhou and Xu Chen. 2019. On-demand Privacy Preservation for Cost-Efficient Edge Intelligence Model Training. In *Provable Security*, Ron Steinfeld and Tsz Hon Yuen (Eds.). Vol. 11821. Springer International Publishing, Cham, 321–329. https://doi.org/10.1007/978-3-030-31919-9_19

Received April 2020; revised July 2020; accepted October 2020