5G-Enabled UAV-to-Community Offloading: Joint Trajectory Design and Task Scheduling

Zhaolong Ning, *Senior Member, IEEE*, Peiran Dong, *Graduate Student Member, IEEE*, Miaowen Wen, *Senior Member, IEEE*, Xiaojie Wang, Lei Guo, Ricky Y. K. Kwok, *Fellow, IEEE*, and H. Vincent Poor, *Life Fellow, IEEE*

***Abstract*— Due to line-of-sight communication links and distributed deployment, Unmanned Aerial Vehicles (UAVs) have attracted substantial interest in agile Mobile Edge Computing (MEC) service provision. In this paper, by clustering multiple users into independent communities based on their geographic locations, we design a 5G-enabled UAV-to-community offloading system. A system throughput maximization problem is formulated, subjected to the transmission rate, atomicity of tasks and speed of UAVs. By relaxing the transmission rate constraint, the mixed integer non-linear program is transformed into two subproblems. We first develop an average throughput maximization-based auction algorithm to determine the trajectory of UAVs, where a community-based latency approximation algorithm is developed to regulate the designed auction bidding. Then, a dynamic task admission algorithm is proposed to solve the task scheduling subproblem within one community. Performance analyses demonstrate that our designed auction bidding can guarantee user truthfulness, and can be fulfilled in polynomial time. Extensive simulations based on real-world data in health monitoring and online YouTube video services show that our proposed algorithm is able to maximize the system throughput while guaranteeing the fraction of served users.**

***Index Terms*— UAV, mobile edge computing, trajectory design, task scheduling, 5G communications.**

Manuscript received October 8, 2020; revised February 19, 2021; accepted April 12, 2021. Date of publication June 14, 2021; date of current version October 18, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFE0206800; in part by the Natural Science Foundation of China under Grant 62025105, Grant 61971084, and Grant 62001073; in part by the Chongqing Talent Program under Grant CQYC2020058659; in part by the National Natural Science Foundation of Chongqing under Grant cstc2019jcyjmsxmX0208; in part by the Fundamental Research Funds for the Central Universities under Grant 2019SJ02; and in part by the U.S. National Science Foundation under Grant CCF-1908308. *(Corresponding authors: Xiaojie Wang; Miaowen Wen.)*

Zhaolong Ning, Xiaojie Wang, and Lei Guo are with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: z.ning@ieee.org; xiaojie.kara.wang@ieee.org; guolei@cqupt.edu.cn).

Peiran Dong is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, and also with the School of Software, Dalian University of Technology, Dalian 116620, China (e-mail: peiran_dong@outlook.com).

Miaowen Wen is with the School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, China (e-mail: eemwwen@scut.edu.cn).

Ricky Y. K. Kwok is with the School of Science and Technology, The Open University of Hong Kong, Hong Kong (e-mail: kwok@ouhk.edu.hk).

H. Vincent Poor is with the Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: poor@princeton.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSAC.2021.3088663.

Digital Object Identifier 10.1109/JSAC.2021.3088663

I. Introduction

OVER the past two decades, Unmanned Aerial Vehicles (UAVs) have experienced a rapid evolution, from primarily military applications to various civilian applications, such as emergency response, disaster management and content delivery [1], [2]. However, these fast-growing applications are generally time-sensitive and require large amounts of computational resources. Diverse promising paradigms, e.g., Mobile Cloud Computing (MCC) and Mobile Edge Computing (MEC), have emerged to bring resources of remote servers to the edge of networks. Traditional Base Station (BS)-based MEC accelerates the progress of complicated applications in many fields, including computation, communication and caching. However, three significant challenges of BS-based MEC exist. First, with the increasing number of Internet-of-Things (IoT) devices, wireless channel resources are becoming scarcer. Second, the location of a BS is relatively fixed, and the limited coverage range of its wireless communication cannot satisfy the demands of numerous users. Third, application demands, such as health monitoring and online video services, are latency-sensitive. A centralized BS can be overloaded when it copes with the requirements of too many users [3]. An on-demand offloading service is required to relieve such burdens on the BS.

In order to address those challenges, UAVs are integrated with MEC to provide pervasive edge computing services for users through computation offloading, e.g., medical information analysis and online video conversion. In addition, the Fifth Generation (5G), as well as several effective spectrum sharing technologies, is developed to enable cellular communications between UAVs and users.

The recent pandemic of Corona Virus Disease 2019 (COVID-19) has caused huge life and economic losses. Its characteristics of extremely contagious and long incubation period have brought significant challenges to traditional healthcare systems. The direct contact between medical staff and patients increases the risk of infections, and surging patients can overload healthcare infrastructure. In addition, patients with chronic diseases need long-term continuous healthcare monitoring. Frequent medical treatment can cause unnecessary monetary expenditure and waste of medical resources. Therefore, a remote eHealth system [4] is needed to facilitate effective and safe treatments. By leveraging smart sensors, patients can be quarantined (if necessary) and monitored through UAVs, which are able to provide sufficient resources for sensors to process the monitored raw data. Then, the analyzed result is transmitted to the healthcare center, e.g., hospitals.

Users in the above scenario can be naturally motivated to form communities for two reasons. First, users in similar geographic locations may have the same kind of tasks. For example, COVID-19 patients in the same quarantine area demand similar health monitoring services. They can be clustered naturally according to their task types and geographic locations. Second, since UAVs provide edge computing services with the objective of maximizing the system throughput, users can be motivated to form groups to contend for the computing resources with strong competitiveness.

As a desirable candidate to extend the capabilities of traditional BSs, UAV-based MEC has advantages in three aspects:

- *Line-of-Sight (LoS) communication links* [5]: Due to the mobility of users, the communication between users and the BS may be interrupted. Although emerging techniques, such as dynamic service migration [6], have been developed to resolve such problems, the migration cost puts extra burdens on the system. With high mobility, UAVs can enable reliable and long-time transmissions by adjusting their locations dynamically.
- *Distributed deployment*: Since the distribution of users is difficult to predict, traditional BSs are empirically deployed, resulting in unbalanced network loads. Conversely, the deployment of UAVs is flexible and not restricted by roads and buildings. In addition, the deployment of UAVs is decentralized, and offloading scheduling can be implemented in a distributed manner.
- *Agile MEC service provision*: With the advantages of high mobility and self-organization, UAVs can provide agile edge computing services [7]. For example, the dynamic deployment of UAVs enables the government to cope with disasters such as earthquakes, which require abundant edge computing resources on a short term basis. In particular, when a sudden outbreak of an infectious disease happens, e.g., COVID-2019, UAVs can be accurately scheduled to patient isolation areas for health monitoring and online treatment services.

Nowadays, many crowd gathering areas, such as cinemas, concerts, shopping malls and train stations, are task-intensive, demanding MEC support. Traditional BSs have difficulty in accommodating numerous users, due to overloaded or limited wireless coverage areas as well as limited spectrum resources. To facilitate the trajectory design of UAVs, we divide users into groups according to their geographic locations, known as user communities. The locations of users within one community are close enough, and existing clustering methods, such as [8], can be leveraged for community division.

Flying without human pilots enables UAVs to provide pervasive MEC services. To guarantee the Quality of Service (QoS) of users, trajectories of UAVs need to be carefully designed, and tasks of users are required to be well scheduled. Recently, some researches addressing on UAV-based trajectory design and task scheduling have been conducted. Zeng and Zhang [9] studied energy-efficient UAV communications by trajectory design. The performance metric related to energy efficiency is defined based on the velocity and acceleration of the UAV. It is an early effort to analyze the relationship between energy consumption and the velocity and acceleration of the UAV. A linear state-space approximation mechanism was proposed to decide the near-optimal trajectory of the UAV. Cheng *et al.* [10] applied UAVs to support data offloading, due to the resource limitations of static BSs. Aiming at maximizing the uploading rate of UAV-served users, an iterative heuristic algorithm was developed to solve the formulated non-convex optimization problem. Considering an existing passive eavesdropper, Ali *et al.* [11] employed *M/M/*1 queueing theory to model cellular and WiFi users. By considering the queueing delay constraint of WiFi users, the optimization problem aimed at minimizing the average queueing latency of cellular users. Block coordinate descent and successive convex approximation algorithms were utilized to solve the formulated mixed-integer non-linear program problem. Xu *et al.* [12] investigated the issue of UAV-enabled wireless power transfer through trajectory design. In order to maximize the received energy, the authors proved that the UAV needs to hover at a fixed location during the transfer period. Furthermore, the fixed location may result in unfair performance among energy receivers. The sum-energy maximization problem is transformed into an alternative problem, aiming at maximizing the minimum energy delivered to all receivers. By relaxing the problem and ignoring the UAV speed constraints, multiple fixed locations can be derived, and the UAV hovers at each location for a certain amount of time.

Although UAVs can provide satisfactory services by designing trajectories and scheduling tasks, spectrum scarcity is still an important obstacle that prevents further development of UAV-assisted MEC services. To overcome this challenge, a few studies integrate UAVs into cellular networks, e.g., 5G networks. Specifically, Non-Orthogonal Multiple Access (NOMA) can be exploited to improve the spectral efficiency. Zhang *et al.* [13] designed a cooperative UAV communication scheme to allocate subchannels for data collection. They focused on maximizing the uplink transmission rate by tuning the speed of UAVs. An iterative heuristic solution was developed to solve the subchannel allocation and speed determination problem. Gapeyenko *et al.* [14] exploited millimeter wave to satisfy capacity and scalability demands of 5G networks. Their method mitigated the influence of occlusions on terrestrial links, and allowed UAVs to route dynamically. Rupasinghe *et al.* [15] deployed NOMA-based UAV communication networks to provide services over a densely packed user area. They proposed a beam scanning method to cover the area completely at a specific altitude. Wu *et al.* [16] proposed a cooperative multi-UAV assisted communication framework, where both trajectory and transmission power of UAVs are optimized to increase the network throughput. A systematic initialization scheme was proposed to speed up the algorithm's convergence.

Although trajectory design and task offloading have been extensively investigated in previous studies, few researches consider joint trajectory design and asynchronous task scheduling. In addition, valid offloading (i.e., the whole offloading process rather than the uplink transmission and computation process) needs to be further studied. In this paper, we design a 5G-enabled UAV-to-community offloading system. Especially, we consider several independent user communities requiring computation support. The UAV can pass through those communities, and provide MEC services for task offloading. We formulate the system throughput maximization problem, subjected to the transmission rate, atomicity of tasks and the speed of the UAV. A joint trajectory design and task scheduling algorithm is proposed to select the target community, and allocate computational resources to users within the selected community. The main contributions are summarized as follows:

- We construct a 5G-enabled UAV-to-community offloading system, with the objective of maximizing the system throughput. Considering the dependence of users within one community and independence among different communities, variables are coupled, and the formulated problem is a Mixed Integer Non-Linear Program (MINLP). Furthermore, the MINLP problem is transformed into two subproblems, by decoupling the variables of trajectory design and task scheduling decision.
- By relaxing the transmission rate constraint, we propose a community-based latency approximation algorithm based on a piecewise function. In addition, an average throughput maximization-based auction algorithm is developed to solve the trajectory design subproblem. The presented auction bidding are able to guarantee user truthfulness.
- A dynamic task admission algorithm is proposed to solve the task scheduling subproblem within one community, where constraints of transmission rate and tasks' atomicity are satisfied. Its time complexity is a quadratic function of the number of users. By properly dividing user communities, the proposed algorithm is feasible in practice.
- Extensive simulations based on real health monitoring and online YouTube video service datasets are conducted to evaluate the effectiveness and efficiency of our proposed algorithms. Numerical results show that our proposed algorithm can maximize the system throughput while guaranteeing the fraction of served users, and an appropriate trade-off between the mobility efficiency and the system throughput can be made.

The rest of this paper is organized as follows. Section II illustrates the system model, and the optimization problem is formulated in Section III. The joint trajectory design and task scheduling algorithm is presented in Section IV. Performance evaluations are illustrated in Section V, followed by the conclusion in Section VI.

II. System Model

Fig. 1 shows an urban environment with one UAV and a set of heterogeneous communities, indexed by $\mathcal{K} = \{1, 2, \cdots, K\}$, where each community consists of N independent users. Due to the overload of base stations or emergencies such as pandemic of diseases, UAVs are demanded by user communities to provide various kinds of MEC services, such as massive data processing and healthcare monitoring.

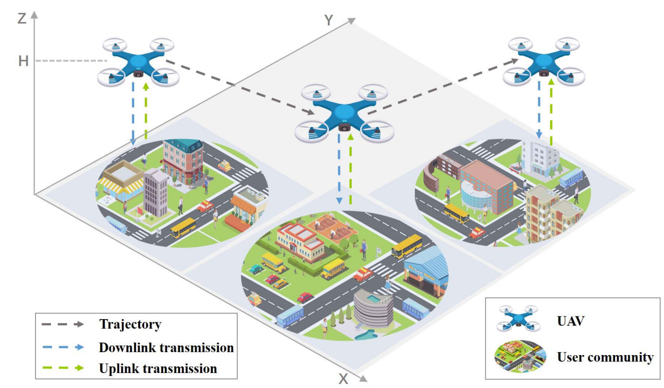


Fig. 1. An illustration of the UAV-to-community offloading scenario.

We design a UAV-assisted offloading system, where the system time horizon is divided into T time slots. The length of each time slot is represented by a small constant $\widetilde{t}$, during which the position of the UAV can be approximately regarded as unchanged. We assume that the UAV is deployed at a fixed altitude H, equaling to the minimum clearance of buildings and terrain avoidance as in [11]. as in [11]. Let vector $\mathbf{p}_t \in \mathbb{R}^{2\times 1}$ denote t. Correspondingly, the horizontal coordinate of user community $k \in \mathcal{K}$ is denoted by vector $\mathbf{q}_k \in \mathbb{R}^{2\times 1}$. We assume that the locations of user communities do not change within the considered time period T, and the coordinate of user $i \in \{1, 2, \cdots, N\}$ in community k can be approximated as $\boldsymbol{q}_k$. Consequently, the communication distance between the UAV and community k in time slot t can be computed by $d_{k,t} = \sqrt{\|\mathbf{p}_t - \mathbf{q}_k\|^2 + H^2}$. To schedule the offloading resources as well as UAV's trajectory (i.e., the service order of user communities), a quasi-static network is considered as in [17], where all user communities generate their tasks at the beginning of each time period. The number of communities remains stable during one system time period, while it can vary across continuous periods. Based on this, we investigate resource allocation of the UAV by taking user community k as an example. The latency-sensitive task generated by user i in community k can be represented by a 4-tuple notion $\langle I_{i,k}, O_{i,k}, \gamma_{i,k}, \tau_{i,k}\rangle$, where $I_{i,k}$ and $O_{i,k}$ denote the input and output data sizes (in bits) of the task, respectively. The ratio of $I_{i,k}$ and $O_{i,k}$ is dependent on tasks, which is high for compression tasks and low for VR/AR applications. Symbol $\gamma_{i,k}$ denotes the computational intensity (in CPU cycles per bit), i.e., $I_{i,k}\gamma_{i,k}$ represents the total required CPU cycles to accomplish the task. Variable $\tau_{i,k}$ records the task deadline, beyond which the results are invalid for user i.

Additionally, the UAV is assumed to work in a full-duplex manner, i.e., task uploading and result downloading are carried out simultaneously. To utilize communication channels efficiently and improve the uploading rate, NOMA technique is leveraged for uplink transmission. Since the input data size is heterogeneous for different tasks, the accomplished time of each task is distinct. Thus, the downlink transmission is asynchronous for different users in one community. In addition, the data size of the results is principally much smaller than that

TABLE I
MAIN NOTATIONS

Notation	Definition
T	System time horizon
$\widetilde{t}$	Length of each time slot
H	Altitude of the UAV
$\mathbf{p}_t$	Horizontal coordinate of the UAV in time slot t
$\mathbf{q}_k$	Horizontal coordinate of user community k
N	Number of users in one community
K	Number of communities
$I_{i,k}$	Input data size of user i in community k
$O_{i,k}$	Output data size of user i in community k
$g_{k,t}$	Channel power gain between the UAV and community k
F	CPU frequency of the UAV

of the uploaded tasks. Thus, Orthogonal Frequency Division Multiple Access (OFDMA) technique is utilized to send results back to users [18]. In this paper, we focus on task scheduling and trajectory design, ad our proposed model is compatible with existing interference cancellation methods such as SIC [19]. The destination node (i.e., the UAV) can perform SIC to reduce the interference from other SD pairs with a smaller equivalent channel gain.

In addition, in order to improve the spectrum efficiency within one community, several devices access the UAV through NOMA simultaneously. For these devices, the start time of task uploading is synchronous and controllable, facilitating the implement of channel multiplexing. By contrast, the start time of task downloading is asynchronous due to different uploading and processing latency. At this time, the UAV cannot provide services for devices in other communities. Each community is served independently and distributedly. Thus, there is no interference from other clusters.

There are three steps for the UAV-based offloading process: (i) UAV admits several users to upload their tasks via NOMA-based uplink transmission; (ii) UAV processes the received tasks; (iii) Results are sent back to users via OFDMA-based downlink transmission. Main notations are summarized in Table I.

A. Communication Model

Considering the free-space path loss and randomness effects in practice, the channel power gain between the UAV and community k in time slot t is typically modeled by [11]:

$$g_{k,t} = \beta_0 \ d_{k,t}^{-2} = \frac{\beta_0}{\|\mathbf{p}_t - \mathbf{q}_k\|^2 + H^2}, \tag{1}$$

where β_0 represents the channel power gain at the reference distance of 1 meter. Let binary variable $a_{i,k,t}$ represent the admission control of task uploading for user i in community k in time slot t, where $a_{i,k,t} = 1$ indicates user i is admitted to upload its task in time slot t; otherwise $a_{i,k,t} = 0$. The achievable uploading rate between user i and the UAV can be calculated by:

$$R^u_{i,k,t} = a_{i,k,t} B \log_2 \left(1 + \frac{P_{i,k} g_{k,t}}{\sum_{j\in\mathcal{N}\setminus\{i\}:a_{j,k,t}=a_{i,k,t}} P_{j,k} g_{k,t} + \sigma^2}\right), \tag{2}$$

where B is the wireless channel bandwidth. Variable $P_{i,k}$ represents the transmission power of user i in community k. Symbol σ^2 is the noise power. We can observe that the uploading rate mainly relies on the channel bandwidth and Signal to Interference plus Noise Ratio (SINR). On the one hand, the channel multiplexing can improve the spectrum efficiency. On the other hand, excessive users sharing one channel may suffer from severe interferences, incurring low transmission rates. Thus, there is a trade-off between the spectrum efficiency and the uplink rate.

Based on OFDMA technique, the interference among users can be omitted. The downloading rate between user i and the UAV can be computed by:

$$R^d_{i,k,t} = B \log_2 \left(1 + \frac{P g_{k,t}}{\sigma^2}\right), \tag{3}$$

where P represents the transmission power of the UAV. Note that the downlink rate does not contain decision variables. We consider a consecutive offloading process, where result backhaul transmission starts immediately when the task is completed. Thus, we mainly focus on the task admission decision.

B. Computational Model

Although tasks are uploaded simultaneously, the received time of each task is different due to heterogeneous data sizes and transmission rates. Traditional sequential processing incurs additional queue latency. To capture the processing time of each task, UAVs can provide parallel edge computing services for multiple users simultaneously by processor sharing [20].

Equipped with an MEC server, the maximum CPU frequency of the UAV is denoted by F (in CPU cycles per second). During time slot t, the total workload accomplished by the UAV can be obtained by $F\widetilde{t}$. Specifically, the completed data size of user i in community k can be computed by:

$$D_{i,k,t} = \frac{F\widetilde{t}}{\mathbf{n}_t}, \tag{4}$$

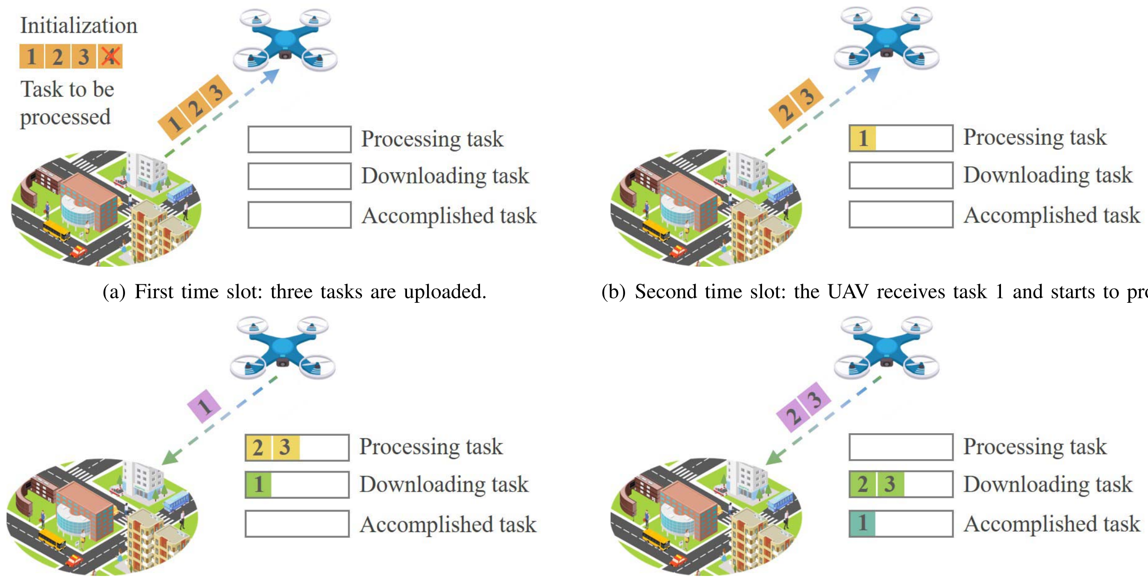
where $\mathbf{n}_t$ records the number of tasks sharing the processor in time slot t. It can be observed that $D_{i,k,t}$ is independent of UAV positions and user communities. UAV can measure the processing time by scheduling the number of task uploading admissions.

III. Problem Formulation

In this section, we analyze constraints of valid offloading, and formulate the system throughput maximization problem.

A. Problem Overview

The offloading process is discussed by an example illustrated in Fig. 2. The four subfigures show an example of task scheduling process in one user community during four successive time slots. In the first time slot, tasks 1-3 are admitted to upload, and task 4 is forbidden due to its long execution time. In the second time slot, the UAV receives task 1 via uplink, and starts to process it. In the third time slot, the UAV



(a) First time slot: three tasks are uploaded. (b) Second time slot: the UAV receives task 1 and starts to process.

(c) Third time slot: Task 1 is downloaded, and other two tasks are processed. (d) Fourth time slot: Task 1 is accomplished, and other two tasks are downloaded.

Fig. 2. An example of the UAV-to-community offloading in four successive time slots.

processes tasks 2-3, and returns the results of task 1 via downlink. In the fourth time slot, task 1 is accomplished. The results of tasks 2-3 are downloaded by the corresponding users. Task 4 can be re-offloaded to the base station or normal edge servers. In reality, it is difficult for resource-constrained UAVs to accomplish all tasks. Our proposed framework focuses on improving the resource utilization efficiency and maximizing the system throughput. However, in order to provide MEC services for all users, the cooperation of the base station or edge servers is indispensable. Similar research can be referred to [21], [22].

Our optimization objective is to maximize the system throughput. Traditional throughput can be calculated by the accumulation of input and output data sizes. However, tasks are accomplished only when the results are downloaded completely, and the tasks exceeding the deadlines are invalid for users. Thus, the system throughput in our constructed UAV-based offloading framework can be measured by the output data size of the accomplished tasks (also called valid data).

B. Constraint Analysis

According to the definition of valid data, the UAV does not process or transmit tasks exceeding the deadline. Since all users generate tasks at the beginning of the system time period, denoted by $t = 0$, its length is set to the minimum deadline for all users, i.e., $\tau^{min} = \min\{\tau_{i,k} | \forall i \in \mathcal{N}, \forall k \in \mathcal{K}\}$. Then, any task accomplished within the system time period satisfies the corresponding deadline. The number of time slots T can be obtained by:

$$T = \lfloor \frac{\tau^{min}}{\widetilde{t}} \rfloor. \tag{5}$$

In addition, due to the limitation of channel conditions, both uplink and downlink communications cannot exceed the maximum achievable transmission rate, i.e.,

$$I_{i,k,t} \leq R^{u}_{i,k,t}\widetilde{t}, \forall i \in \mathcal{N}, \forall k \in \mathcal{K}, \forall t \in [0,T], \tag{6}$$

$$O_{i,k,t} \leq R^{d}_{i,k,t}\widetilde{t}, \forall i \in \mathcal{N}, \forall k \in \mathcal{K}, \forall t \in [0,T], \tag{7}$$

where variables $I_{i,k,t}$ and $O_{i,k,t}$ denote the size of uploading and downloading data in time slot t, respectively. Note that for uplink transmission, the task needs to be admitted to upload. For example, when admission variable $a_{i,k,t} = 0$, channel resources are not available to user i in community k in time slot t, i.e., $I_{i,k,t} = 0$. By utilizing the "big-*M*" method in [23], the uplink admission constraint can be yielded as:

$$I_{i,k,t} \leq M a_{i,k,t}, \forall i \in \mathcal{N}, \forall k \in \mathcal{K}, \forall t \in [0,T], \tag{8}$$

where M is a sufficiently large constant. When $a_{i,k,t} = 1$, user i is admitted to upload its task in time slot t. Since coefficient M is larger than any possible value that may appear on the left side of the inequality, this constraint becomes redundant. Otherwise, $I_{i,k,t}$ is enforced to be 0 when $a_{i,k,t} = 0$.

We consider that tasks cannot be further compressed or partitioned, i.e., the UAV can merely decide whether to accomplish the whole task or reject the task requests. To guarantee the completeness and atomicity of tasks, numerical relationships can be formulated as:

$$\sum_{t=0}^{T} I_{i,k,t} \in \{0, I_{i,k}\}, \forall i \in \mathcal{N}, \forall k \in \mathcal{K}, \tag{9}$$

$$\sum_{t=0}^{T} D_{i,k,t} \in \{0, D_{i,k}\}, \forall i \in \mathcal{N}, \forall k \in \mathcal{K}, \tag{10}$$

$$\sum_{t=0}^{T} O_{i,k,t} \in \{0, O_{i,k}\}, \forall i \in \mathcal{N}, \forall k \in \mathcal{K}, \tag{11}$$

where constraints (9) and (10) guarantee the atomicity of tasks. Constraint (11) ensures the whole results are sent back to the corresponding users and the offloading tasks are valid.

In addition, due to the limitation of physical equipment's, the maximum speed of the UAV is denoted by v^{max}. For simplicity, we do not consider vector velocity and acceleration of the UAV. The position constraint across successive time slots can be written by:

$$\|\mathbf{p}_{t+1} - \mathbf{p}_t\| \leq v^{max}\widetilde{t}, \forall t \in [0, T-1]. \tag{12}$$

C. Problem Formulation

Based on the analyzed constraints, the system throughput maximization problem is formulated as follows:

$$\mathbf{P1}: \max_{\mathbf{p}_t, a_{i,j,k}} \sum_{t=0}^{T}\sum_{k=1}^{K}\sum_{i=1}^{N} O_{i,k,t}, \quad \text{s.t.} \quad \text{Constraints (6)-(12).} \tag{13}$$

Due to the mobility of the UAV, the significant challenge to solve problem $\mathbf{P1}$ is the non-linear transmission rate in constraints (6) and (7). Besides, norm calculation in constraint (12) makes it a quadratic constraint. Both admission variables $a_{i,k,t}$ and constraints (9)-(11) are binary. We can observe that constraints (8) and (12) directly depend on decision variables $a_{i,j,k}$ and $\mathbf{p}_t$, respectively. The position variable $\mathbf{p}_t$ determines the communication distance between the UAV and the community. Furthermore, the communication distance affects the communication rate (including both uplink and downlink). Thus, the position variable $\mathbf{p}_t$ and constraints (6)-(7) are inter-dependent. In addition, since each task is uploaded, processed and downloaded sequentially, constraints of task processing and downloading (equations (10) and (11)) are dependent on the constraint of task uploading (equation (9)). In particular, in constraints (6) and (7), the left side of the inequalities are affected by $a_{i,j,k}$, while $\mathbf{p}_t$ influences that of the right side. These two decision variables are coupled, making the formulated problem difficult to solve. Accordingly, problem $\mathbf{P1}$ is an MINLP, which is more complicated than the 0-1 knapsack problem [24]. It is challenging to obtain the optimal solution by the algorithms with polynomial time complexity. Therefore, problem $\mathbf{P1}$ needs to be relaxed into Mixed Integer Linear Program (MILP).

D. Problem Transformation

Decision variables $\mathbf{p}_t$ and $a_{i,k,t}$ are coupled in constraint (6), making it complex to obtain the optimal scheduling. To decouple the decision variables and resolve problem $\mathbf{P1}$, we relax constraints (6) and (9)-(11) by allowing all users in one community to upload their tasks simultaneously, i.e., $a_{i,k,t}$ is removed from the uploading transmission rate in equation (2). In addition, we define a piecewise function $L_{i,k}(\cdot)$ to approximate the task execution latency (including uploading, processing and downloading latencies) of user i in community k, where $L_{i,k}(\cdot)$ is the function of the uplink and downlink rates. Since all tasks are latency-sensitive and their deadlines cannot be exceeded, the latency constraint can be reformulated as:

$$L_{i,k}(R^u_{i,k,t}, R^d_{i,k,t}) \leq \tau_{i,k}, \forall i \in \mathcal{N}, \forall k \in \mathcal{K}. \tag{14}$$

Then, problem $\mathbf{P1}$ is transformed into:

$$\mathbf{P2}: \max_{\mathbf{p}_t} \sum_{t=0}^{T}\sum_{k=1}^{K}\sum_{i=1}^{N} O_{i,k,t}, \quad \text{s.t. Constraints (6), (7), (12), (14).} \tag{15}$$

Note that we solve problem $\mathbf{P1}$ in two steps by decoupling two decision variables: (i) We select the trajectory of the UAV by solving problem $\mathbf{P2}$. A community-based latency approximation algorithm is proposed to fit the piecewise function $L_{i,k}(\cdot)$. Then, an auction-based algorithm is developed to determine the service order of all communities; (ii) The following formulated problem $\mathbf{P3}$ is resolved to maximize the throughput within one community:

$$\mathbf{P3}: \max_{a_{i,j,k}} \sum_{t=0}^{T}\sum_{i=1}^{N} O_{i,k,t}, \forall k \in \mathcal{K}, \quad \text{s.t. Constraints (6)-(11).} \tag{16}$$

Given the trajectory of the UAV, problem $\mathbf{P3}$ is much easier than $\mathbf{P1}$, and we develop a dynamic task admission algorithm to schedule the tasks of users within the selected community.

IV. Joint Trajectory Design and Task Scheduling

In this section, we propose the joint Trajectory Design and Task Scheduling (TDTS) algorithm to solve problem $\mathbf{P1}$. Specifically, TDTS algorithm consists of three sub-algorithms, where community-based latency approximation and average throughput maximization-based auction algorithms are developed to solve problem $\mathbf{P2}$. Then, a dynamic task admission algorithm is designed to solve problem $\mathbf{P3}$.

A. Community-Based Latency Approximation

Many existing researches have investigated the time approximation methods, such as Sample Average Approximation (SAA) [25] and 2-D grid-based linear approximation [17]. SAA method conducts Monte-Carlo sampling by allowing the UAV to move randomly among communities. However, it is a low-efficient scheduling strategy since the communication stability between the UAV and users cannot be guaranteed. The 2-D grid-based linear approximation is suitable for situations where users are dispersed. Considering the community characteristics of users, we propose a community-based latency approximation algorithm.

Given an initial position $\mathbf{p}_0$ of the UAV, the total task execution latency of all users in community k can be approximated by assuming that the UAV moves straight forward to the location of community k with maximum speed v^{max}, until it reaches right above community k, i.e.,

$$\begin{aligned} &\|\mathbf{p}_{t+1} - \mathbf{p}_t\| = v^{max}\widetilde{t}, && \text{if } \mathbf{p}_t \neq \mathbf{q}_k; \\ &\|\mathbf{p}_{t+1} - \mathbf{p}_t\| = 0, && \text{if } \mathbf{p}_t = \mathbf{q}_k, \end{aligned} \tag{17}$$

where the direction of the UAV can be represented by vector $\vec{\mathbf{p}_t\mathbf{q}_k}$. Obtaining the trajectory scheduling, the communication distance between the UAV and community k across successive time slots can be accurately computed. Since all users in community k are admitted to upload their tasks, the uplink rate of user i in time slot t can be determined by:

$$r^u_{i,k,t} = B\log_2\left(1+\frac{P_{i,k}g_{k,t}}{\sum_{j\in\mathcal{N}\setminus\{i\}}P_{j,k}g_{k,t}+\sigma^2}\right). \tag{18}$$

Without uploading admission variables, the task execution latency merely depends on the position of the UAV. Algorithm 1 presents the pseudo-code of the community-based latency approximation algorithm. It allows the UAV to move straight to each community with the maximum speed from the initial position (line 6). The uplink transmission rate can be computed by relaxing constraints (8)-(11) (line 7). Let t^u, t^p and t^d denote time slots that task uploading, processing and downloading procedures are completed, respectively (lines 9, 14 and 19). Accordingly, the task execution time of each user can be approximated.

Algorithm 1 Community-based Latency Approximation Algorithm

1: **Input:** $\mathbf{p}_0$, $\mathbf{q}_k$, v^{max}, F, $\widetilde{t}$, $\langle I_{i,k}, O_{i,k}, \gamma_{i,k}, \tau_{i,k}\rangle$;
2: **Initialization:** $t=0$, $k=1$, $\{L_{i,k}\}=\mathbf{0}$;
3: **while** $k \le K$ **do**
4: $\mathbf{p}^{ini} = \mathbf{p}_0$;
5: **while** $t \le T$ **do**
6: Allow the UAV to move straight to community k with speed v^{max} from initial position $\mathbf{p}^{ini}$;
7: Calculate $r^u_{i,k,t}$ by equation (18);
8: **if** $\sum_{t'=0}^{t} r^u_{i,k,t'}\widetilde{t} \ge I_{i,k}$ **then**
9: $t^u = t$;
10: $L_{i,k} = L_{i,k} + t^u$;
11: Calculate the task process rate by equation (4);
12: **end if**
13: **if** $\sum_{t'=t^u}^{t} D_{i,k,t'} \ge I_{i,k}$ **then**
14: $t^p = t - t^u$;
15: $L_{i,k} = L_{i,k} + t^p$;
16: Calculate $r^d_{i,k,t}$ rate by equation (3);
17: **end if**
18: **if** $\sum_{t'=t^p}^{t} r^d_{i,k,t'}\widetilde{t} \ge O_{i,k}$ **then**
19: $t^d = t - t^u$;
20: $L_{i,k} = L_{i,k} + t^d$;
21: **end if**
22: Update $\mathbf{p}_{t+1}$ by equation (17);
23: **end while**
24: $t = 0$;
25: **end while**
26: **Return:** $\{L_{i,k}\}$.

B. Throughput Maximization-Based Auction

With the objective of maximizing the system throughput, intuitively, the output data size needs to be maximized and the task execution time needs to be minimized. By considering that user communities compete for edge computing resources of the UAV, we propose an average throughput maximization-based auction algorithm to solve problem $\mathbf{P2}$.

We design the auction bidding of each community by transforming the long-term optimization objective into problem $\mathbf{P2}$. Specifically, we aim to maximize the system throughput during the whole system time period, which is equivalent to maximizing the corresponding average throughput, i.e.,

$$\max\sum_{t=0}^{T}\sum_{k=1}^{K}\sum_{i=1}^{N}O_{i,k,t} \Leftrightarrow \max\sum_{k=1}^{K}\sum_{i=1}^{N}\frac{O_{i,k}}{T}, \tag{19}$$

where the system time period T can be replaced by the effective time (i.e., task execution time) of user i in community k. By executing Algorithm 1, the modified average throughput can be represented by $O_{i,k}/L_{i,k}$. For each task, there is a "compression" ratio of the output and input data size, i.e.,

$$\eta_{i,k} = \frac{O_{i,k}}{I_{i,k}}, \forall i\in\mathcal{N}, \forall k\in\mathcal{K}. \tag{20}$$

In general, the "compression" ratio for one type of application is relatively fixed. Existing tools can easily measure the ratios of all kinds of applications, and we assume that the UAV grasps these ratios in advance. Based on the "compression" ratio, we design the auction bidding for user i in community k as follows:

$$b_{i,k} = \frac{I_{i,k}\eta_{i,k}}{L_{i,k}}, \forall i\in\mathcal{N}, \forall k\in\mathcal{K}. \tag{21}$$

Theorem 1: *Given the designed auction bidding in equation (21), all users are motivated to bid truthfully without malicious misrepresentations.*

Proof: Considering the selfishness of users, they may bid more by exaggerating input data size $I_{i,k}$ to increase the possibility of winning the edge computing resources. The exaggerated bidding of user i in community k can be denoted by:

$$\widehat{b}_{i,k} = \frac{(I_{i,k}+\Delta I_{i,k})\eta_{i,k}}{L_{i,k}+\Delta L_{i,k}}, \forall i\in\mathcal{N}, \forall k\in\mathcal{K}, \tag{22}$$

where $\Delta I_{i,k}$ and $\Delta L_{i,k}$ denote the misrepresented data size and the corresponding extra execution time of user i in community k, respectively. To study the effect of misrepresentation, we keep other variables unchanged. Specifically, let r^u, r^p and r^d represent the constant uplink, processing and downlink rates, respectively. Then, the truthful bidding and the exaggerated bidding of user i can be obtained in (23) and (24), as shown at the bottom of the next page.

It can be observed that when the uplink, processing and downlink rates are fixed, the bidding of user i is independent of its input data size, and $b_{i,k} = \widehat{b}_{i,k}$ can be obtained. The independence between the bidding and the input data size can motivate users to bid truthfully. The proof is completed. □

Based on the designed bidding, the pseudo-code of the average throughput maximization-based auction mechanism is illustrated in Algorithm 2. Considering the continuity of time and space, the position of the UAV varies across successive time slots, and the corresponding parameters change accordingly. For example, in time slot $t = 0$, the UAV selects community k^* by Algorithm 2, where users' task execution time is approximated based on initial position $\mathbf{p}_0$. After the UAV completes the tasks of community k^*, it utilizes Algorithm 2 to choose the next service target. At this time, it is based on the current position of the UAV, i.e., $\mathbf{p}_{k^*}$, so that the task execution time can be approximated.

Algorithm 2 Average Throughput Maximization-based Auction Algorithm

Input: $\mathbf{p}_0$, $\mathbf{q}_k$, v^{max}, F, $\widetilde{t}$, $\langle I_{i,k}, O_{i,k}, \gamma_{i,k}, \tau_{i,k}\rangle$;
Initialization: $i = 1$, $k = 1$, $\{b_{i,k}\} = \mathbf{0}$;
Approximate task execution time by Algorithm 1;
while $k \leq K$ **do**
  while $i \leq N$ **do**
    Calculate $b_{i,k}$ based on equation (21);
  end while
end while
for community k **do**
  Submit the accumulated bidding $\sum_{i=1}^{N} b_{i,k}$;
end for
Determine the bid winner $k^* = \arg\max_k \sum_{i=1}^{N} b_{i,k}$;
Return: k^*.

C. Dynamic Task Admission

After solving problem $\mathbf{P2}$, the trajectory of the UAV can be obtained, and the uplink transmission rate in equation (6) merely depends on task admission variable $a_{i,k,t}$. We can observe that constraints (9)-(11) make problem $\mathbf{P3}$ similar to 0-1 knapsack problem, since each task is either totally uploaded or rejected. Considering the UAV as a knapsack, the optimization objective is to put as many tasks as possible into the knapsack, under the premise that their execution time does not exceed the deadline. However, problem $\mathbf{P3}$ is more complicated than 0-1 knapsack problem for two reasons:

- The threshold in traditional 0-1 knapsack problem is the volumn of the knapsack, which is identical to all "stuffings". However, the threshold in problem $\mathbf{P3}$ is the task deadline, which is related to the characteristics of applications. In such cases, traditional dynamic programming is not suitable for problem $\mathbf{P3}$.
- In traditional 0-1 knapsack problem, all "stuffings" are independent, i.e., their weights are inherent properties. However, aiming at maximizing the average system throughput of users in one community, the weights of tasks (i.e., their contributed throughputs) depend on each other. First, due to the utilization of NOMA technique, interferences from other users decease the uplink rate. Second, multiple tasks share the computational resources of the UAV. The processing latency of each task depends on the number of tasks processed in parallel.

To address the above mentioned challenges, we propose a dynamic task admission algorithm to determine the task uploading order within one community. The corresponding pseudo-code is presented in Algorithm 3. The key idea in traditional dynamic programming for 0-1 knapsack problem is to accept the "stuffings" that can increase the utilization efficiency of unit space. Accordingly, in our proposed dynamic task admission algorithm, tasks that can improve the average throughput are admitted to upload simultaneously (lines 8-17 and lines 20-29). Specifically, let $\mathcal{P}^*$ and $\mathcal{A}$ denote the current maximum throughput and admission set, respectively. Two conditions are considered for task admission scheduling, i.e., admitting task i can improve the current maximum throughput (line 8) or replacing task j in $\mathcal{A}$ by task i can improve the current maximum throughput (line 20). Note that sufficient time should be reserved for task execution (constraints (9)-(11)). Thus, we judge whether admitting task i incurs task timeout or not (lines 11 and 23), where the execution time of all tasks in $\mathcal{A}$ should be less than the remaining time, i.e., $\min\{\tau_{j,k^*}, \text{T-t}\}$.

D. Performance Analysis

With the objective of maximizing system throughput, we construct a dynamic UAV-based offloading framework. Three nested algorithms are proposed to approximate the task execution time, determine the trajectory of the UAV and allocate resources to users for task execution, respectively. The pseudo-code of TDTS algorithm is shown in Algorithm 4.

Theorem 2: *The time complexity of TDTS algorithm is $O(N^2T^2)$, where N and T denote the number of users in one community and that of time slots in one period, respectively.*

Proof: The time complexity of TDTS algorithm mainly depends on Algorithms 1-3.

According to the pseudo-code of Algorithm 1, its time complexity is $O(KT)$, where K and T denote the number of user communities and that of time slots in one system time period, respectively. If Algorithm 1 is applied to approximate task execution latency approximation, the time complexity of Algorithm 2 is $O(K(N+T))$.

Now we analyze the time complexity of Algorithm 3. After deciding the trajectory, the UAV judges all users in the selected community sequentially. In lines 9 and 21, the maximum number of admitted tasks in set $\mathcal{A}$ is $i-1, 1 \leq i \leq N$. In addition, in lines 10 and 22, the time complexity of calculating the task execution time is equivalent to that of Algorithm 1. Therefore, the total time complexity of Algorithm 3 can be represented by $O(\frac{1}{2}((N-1)NT+2)K(N+T))$. For simplicity, we omit constant low-order terms. The time complexity of Algorithm 3 is $(N^2KT(N+T))$. In general, the number of communities is much less than that of users, i.e., $K \ll N$. Thus, it can be obtained that the time complexity of Algorithm 3 is $O(N^2T(N+T))$, where N denotes the number of users in one community. Accordingly, the time complexity of TDTS algorithm can be represented by $\max\{O(N^3T), O(N^2T^2)\} = O(N^2T^2)$, and the number of users is much smaller than that of time slots, i.e. $N \ll T$.

$$b_{i,k} = \frac{I_{i,k}\eta_{i,k}}{I_{i,k}/r^u + I_{i,k}/r^p + I_{i,k}\eta_{i,k}/r^d}, \forall i \in \mathcal{N}, \tag{23}$$

$$\widehat{b}_{i,k} = \frac{(I_{i,k}+\Delta I_{i,k})\eta_{i,k}}{(I_{i,k}+\Delta I_{i,k})/r^u + (I_{i,k}+\Delta I_{i,k})/r^p + (I_{i,k}+\Delta I_{i,k})\eta_{i,k}/r^d}. \tag{24}$$

Algorithm 3 Dynamic Task Admission Algorithm

Input: $\mathbf{p}_0$, $\mathbf{q}_k$, v^{max}, F, $\widetilde{t}$, $\langle I_{i,k}, O_{i,k}, \gamma_{i,k}, \tau_{i,k} \rangle$;
Initialization: $t = 0$, $i = 1$, $\{a_{i,k,t}\} = \mathbf{0}$;
Select community k^* based on Algorithm 2;
while $t \leq T$ **do**
  Admission set $\mathcal{A} = \varnothing$, $\mathcal{P}^*(\mathcal{A}) = 0$;
  for user i in community k^* **do**
    Calculate $\mathcal{P}^*$ based on $\mathcal{A} \cup \{i\}$;
    if $\mathcal{P}^*(\mathcal{A} \cup \{i\}) \geq \mathcal{P}^*(\mathcal{A})$ **then**
      for $j \in \mathcal{A} \cup \{i\}$ **do**
        Calculate task execution time l_{j,k^*};
        if $l_{j,k^*} > \min\{\tau_{j,k^*}, \text{T-t}\}$ **then**
          $a_{i,k,t} = 0$;
          continue;
        end if
      end for
      $a_{i,k,t} = 1$;
      $\mathcal{A} = \mathcal{A} \cup \{i\}$;
    else
      for $j \in \mathcal{A}$ **do**
        if $\mathcal{P}^*((\mathcal{A} \backslash \{j\}) \cup \{i\}) > \mathcal{P}^*(\mathcal{A})$ **then**
          for $s \in (\mathcal{A} \backslash \{j\}) \cup \{i\}$ **do**
            Calculate task execution time l_{s,k^*};
            if $l_{s,k^*} > \min\{\tau_{j,k^*}, \text{T-t}\}$ **then**
              $a_{i,k,t} = 0$;
              continue;
            end if
          end for
          $a_{i,k,t} = 1$, $a_{j,k,t} = 0$;
          $\mathcal{A} = (\mathcal{A} \backslash \{j\}) \cup \{i\}$;
        end if
      end for
    end if
  end for
end while
Return: $\{a_{i,k,t}\}$.

The proof can be completed. □

Theorem 3: *Ideally, without the task deadline constraint, the total average throughput of community k increases with the rising number of admitted tasks, denoted by n^*, in community k. In practice, if and only if the number of admitted tasks satisfies the following inequality, task i in community k is valid*:

$$\frac{I_{i,k}}{1/(n^*+C)} + \frac{I_{i,k}}{F/n^*} + \frac{I_{i,k}\eta_{i,k}}{1/C} \leq \min\{\tau_{i,k}, T\}, \quad \forall i \in \mathcal{N}, \forall k \in \mathcal{K}, \tag{25}$$

where constant $C = \sigma^2/P_{i,k}g_{k,t}$, and $\min\{\tau_{i,k}, T\}$ denotes the remaining time (i.e., deadline constraint) for task i.

Proof: Without the task deadline constraint, all tasks are allowed to transmit and process in parallel. According to equation (2), we can observe that the uplink rate is proportional to the SINR. Correspondingly, the downlink rate is proportional to the Signal to Noise Ratio (SNR). For simplicity, we represent the uplink and downlink rates by SINR and SNR in the following, respectively. In addition, we assume the transmission power of all users is equal to that of the UAV. Since the locations of all users can be approximated by that of community k, the channel power gain can be approximately equal, denoted by g. Based on the designed auction bidding in equation (21), the total average throughput can be represented by:

$$\begin{aligned}
&\sum_{i=1}^{n^*} \frac{I_{i,k}\eta_{i,k}}{\frac{I_{i,k}}{Pg/(xPg+\sigma^2)} + \frac{I_{i,k}}{F/y} + \frac{I_{i,k}\eta_{i,k}}{Pg/\sigma^2}} \\
&\quad = \sum_{i=1}^{n^*} \frac{\eta_{i,k}}{(xPg+\sigma^2)/Pg + y/F + \eta_{i,k}\sigma^2/Pg} \\
&\quad = \sum_{i=1}^{n^*} \frac{\eta_{i,k}}{(\eta_{i,k}+1)C + x + y/F} \\
&\quad \geq \sum_{i=1}^{n^*} \frac{\eta_{i,k}}{(\eta_{i,k}+1)C + (1+1/F)n^*}, \forall k \in \mathcal{K},
\end{aligned} \tag{26}$$

where constant $C = \sigma^2/Pg$. Variables x and y denote the number of tasks transmitted and processed in parallel, where $x \leq n^*$ and $y \leq n^*$ hold.

It can be observed that the auction bidding of user i, i.e., the average throughput, is always positive without the deadline constraint. Thus, ideally, the total average throughput of community k increases with the rising number of admitted tasks. In practice, tasks are time-sensitive, and we merely consider valid tasks during the system time period. Accordingly, the remaining time for task i can be represented by $\min\{\tau_{i,k}, T\}$. If and only if the task execution time does not exceed the deadline constraint, the task is valid, i.e.,

$$\begin{aligned}
&\frac{I_{i,k}}{Pg/(xPg+\sigma^2)} + \frac{I_{i,k}}{F/y} + \frac{I_{i,k}\eta_{i,k}}{Pg/\sigma^2} \\
&\quad = \frac{I_{i,k}}{1/(x+C)} + \frac{I_{i,k}}{F/y} + \frac{I_{i,k}\eta_{i,k}}{1/C} \\
&\quad \leq \frac{I_{i,k}}{1/(n^*+C)} + \frac{I_{i,k}}{F/n^*} + \frac{I_{i,k}\eta_{i,k}}{1/C} \\
&\quad \leq \min\{\tau_{i,k}, T\}.
\end{aligned} \tag{27}$$

The proof can be completed. □

Through **Theorem 3**, we can observe that given the task and communication information in one community, the upper bound of the number of admitted tasks can be derived.

Algorithm 4 Trajectory Design and Task Scheduling Algorithm

Input: $\mathbf{p}_0$, $\mathbf{q}_k$, v^{max}, F, $\widetilde{t}$, $\langle I_{i,k}, O_{i,k}, \gamma_{i,k}, \tau_{i,k} \rangle$;
for each system time period **do**
  Select community k^* by Algorithm 2;
  The UAV moves to the location of community k^*;
  Schedule tasks in community k^* by Algorithm 3;
end for
Return: $\{O_{i,k,t}\}$.

TABLE II
SIMULATION PARAMETERS

Parameter	Value
Number of communities M	5
Channel bandwidth B	10MHz
Maximum UAV speed v^{max}	2 m/time slot
Transmission power of the UAV P	23 dBm
Noise power σ^2	-96 dBm
UAV altitude H	100 m
UAV CPU frequency F	5 GHz or 25 GHz
Length of one time slot $\widetilde{t}$	0.1 s
Reference path loss β_0	-50 dB

V. PERFORMANCE EVALUATION

In this section, extensive simulations are conducted to verify the effectiveness and efficiency of our solution. First, we introduce the simulation setup. Then, numerical results are presented.

A. Simulation Setup

We utilize the Electroencephalography (EEG) dataset in [26] and the online YouTube video service in [27] to evaluate our proposed TDTS algorithm. EEG dataset records the brain state information of patients. Various brain diseases such as epileptic seizures can be monitored by analyzing EEG time series data. Each task for medical analysis consists of 4096 samples of EEG data. The data size of each sample is between [560, 747] KB. The data size of the online YouTube video service varies between [30, 450] MB. According to the statistical results of the YouTube video file size distributions, the data size of most videos is less than 80 MB. In addition, compared with traditional base stations and servers on the ground, UAVs are relatively small in both physical size and storage capacity, which are suitable for processing tasks with relatively small amounts of data. The algorithm is also designed for this purpose. In our simulation, the UAV and user communities are randomly distributed in an area of 2 km $\times$ 2 km. The altitude of UAV is set to 100 m [13]. We consider two scenarios with sufficient and shortage of computational resources, where the CPU frequencies of UAV are set to 5 GHz and 25 GHz, respectively. Simulation parameters are summarized in Table II.

To demonstrate the efficiency and effectiveness of our proposed algorithm, we consider three performance metrics as follows:

- *System throughput*: The total valid output data during the system time period.
- *Service percentage*: The number of users that receive valid MEC services divided by the total number of users in the selected community.
- *Mobility efficiency*: The total valid output data divided by the trajectory length of the UAV (in MB per meter).

Inspired by [17], our proposed algorithm is compared with the following four methods:

- *Static BS (SBS)*: A static BS is deployed at the initial location of the UAV.
- *Maximum Data First (MDF)*: Based on the optimization objective, the UAV always admits the user with the maximum output size within the selected community in priority.
- *Shortest Distance First (SDF)*: To exploit spectrum resources efficiently, the UAV serves the nearest community in priority.
- *Sine Cosine Algorithm (SCA)*: It is a population-based optimization algorithm, which creates multiple initial random candidate solutions, and then utilizes mathematical models based on sine and cosine functions to make these solutions fluctuate towards the optimal direction or in the opposite direction.

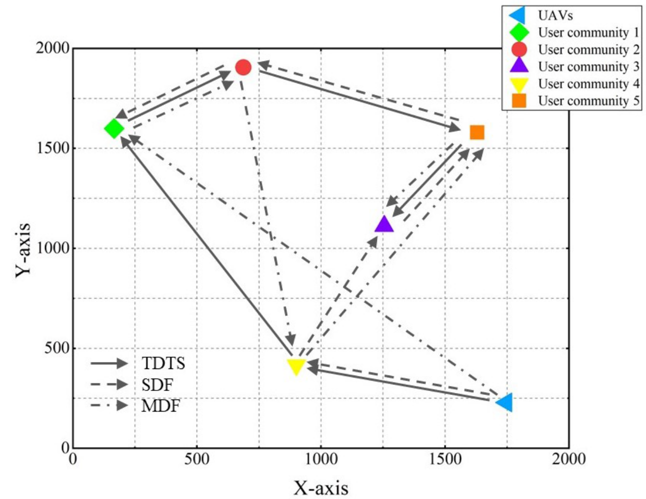
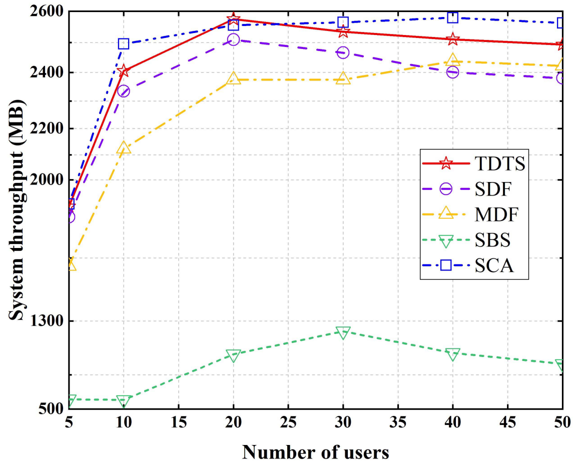


Fig. 3. Illustration of the UAV trajectories based on three algorithms.

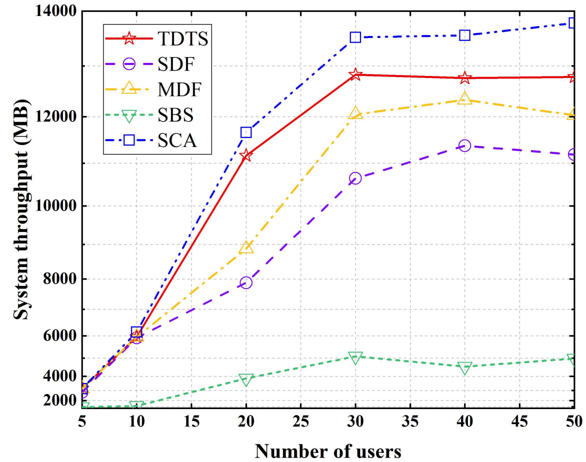
B. Numerical Results

The UAV trajectories based on three algorithms are illustrated in Fig. 3. We can observe that SDF method guides the UAV to follow the shortest path and provide edge computing services. However, it does not consider the requirement of different user communities. For example, in Fig. 3, user community 1 can contribute to the largest system throughput among all the communities. However, it is ranked forth by SDF method, leading to the performance loss. In addition, MDF algorithm pursues short-sighted maximization of the system throughput, causing the UAV to waste a lot of time moving among distributed communities. Different from the above two methods, our proposed TDTS algorithm can reach an appropriate trade-off between the system throughput maximization and the trajectory distance minimization with a satisfactory performance.

The system throughput of four algorithms under different numbers of users is plotted in Fig. 4. When the number of users is relatively small (e.g., $N \leq 20$), the system throughput increases when N grows. It is because more tasks are accomplished by the UAV. However, when N continues to grow (e.g., $N > 20$), the system throughput remains relatively stable and does not increase further. This is because computational resources in shortage limit the service capability of the UAV. Only part of tasks are admitted to be uploaded. While in a resource-sufficient scenario, all tasks are processed by the

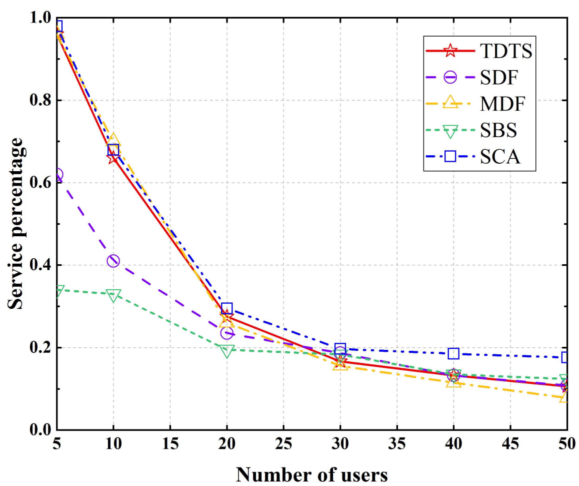


(a) F = 5 GHz, T = 40000 slots

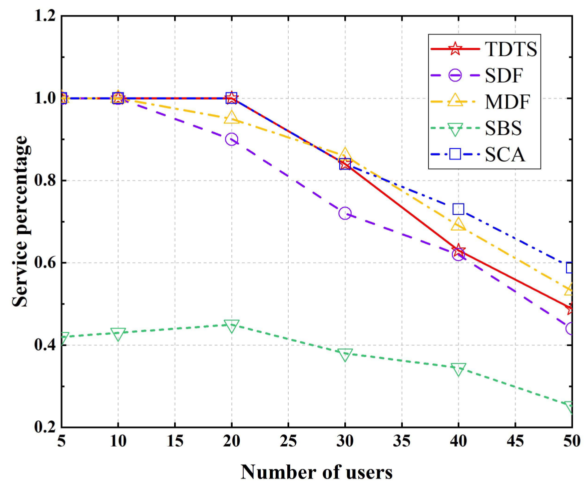


(b) F = 25 GHz, T = 40000 slots

Fig. 4. System throughput with different numbers of users.



(a) F = 5 GHz, T = 40000 slots

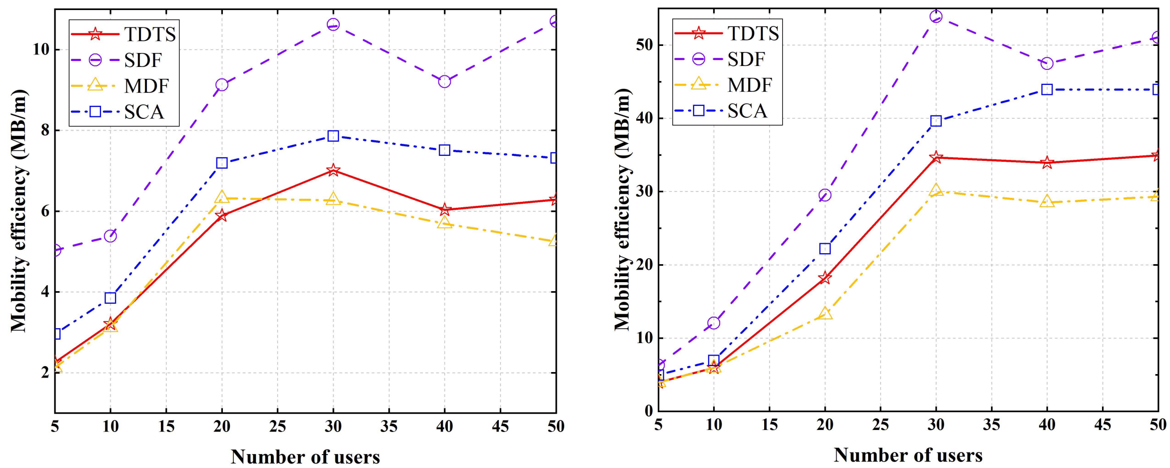


(b) F = 25 GHz, T = 40000 slots

Fig. 5. Service percentage with different numbers of users.

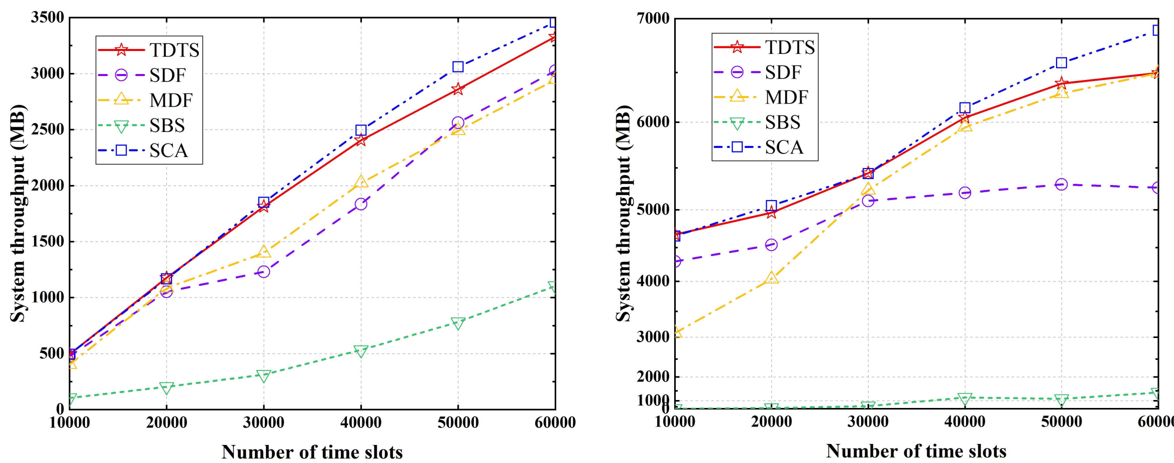
UAV, and the system throughput equals to the total output data size of all users in the selected community. In Fig. 4(a), our proposed TDTS algorithm performs 5%, 8% and 155% better than the compared SDF, MDF and SBS methods, respectively. While in Fig. 4(b), these ratios increase to 18%, 10% and 185%, respectively. This is because when computational resources are in shortage, the UAV cannot accomplish the task with a large data size, and the corresponding options are limited. Although SCA algorithm can improve the performance compared to TDTS algorithm, its convergence time is the cubic function of the system time period. Both TDTS and the compared algorithms admit the task with a relatively small data sizes. The community, where the average data size of users is small, is preferred. Thus, the performance gaps are small. Conversely, TDTS algorithm is able to select those tasks that can improve the average throughput efficiently by dynamic programming, when computational resources are sufficient. In summary, our proposed TDTS algorithm outperforms the compared methods in terms of system throughput. When computational resources are sufficient, the system throughput increases significantly.

In Fig. 5, we focus on the service percentage under different numbers of users, which reflects the fraction of served users. When the number of users N is relatively small (e.g., $N \leq 10$), the UAV is capable to accomplish most tasks during the system time period, and the service percentage is large. With the increase of N ($10 < N < 30$), the UAV becomes fully loaded, and excessive tasks are forbidden to be uploaded. Accordingly, the service percentage drops rapidly. In Fig. 5(a), the downward trend slows down when N is relatively large ($N \geq 30$). The main limiting factor is the computation capability of the UAV, while in Fig. 5(b), the main limiting factor is the number of time slots. Tasks with large data sizes cannot be accomplished during the system time period due to large transmission latency. Our proposed TDTS algorithm has similar performance with MDF and SCA methods, and better than other two schemes. In summary, our proposed TDTS algorithm can maximize the



(a) F = 5 GHz, T = 40000 slots (b) F = 25 GHz, T = 40000 slots

Fig. 6. Mobility efficiency with different numbers of users.



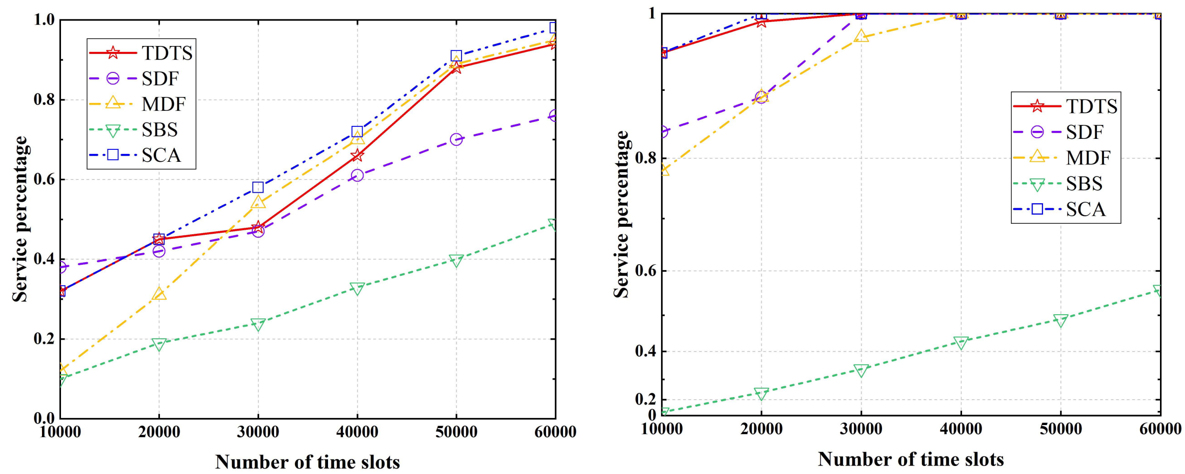
(a) F = 5 GHz, N = 10 (b) F = 25 GHz, N = 10

Fig. 7. System throughput with different numbers of time slots.

system throughput while guaranteeing the fraction of served users.

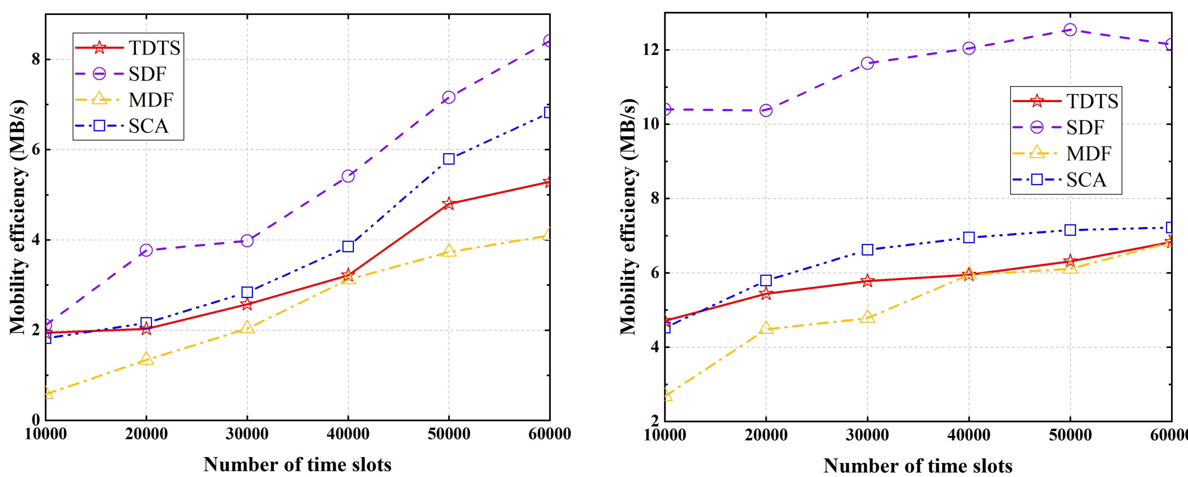
The mobility efficiency of three algorithms under different number of users is presented in Fig. 6. Since BS is deployed statically, we do not consider its mobility efficiency. When the number of users N grows, more tasks are accomplished, and the mobility efficiency increases. SDF and SCA methods choose the nearest user community and adjust the trajectory in each iteration to provide MEC services, respectively, so that their mobility efficiencies are larger than those of TDTS and MDF algorithms. When N becomes large (e.g., $N \geq 30$), the system throughput remains stable. Accordingly, the mobility efficiency changes within a relatively stable range. Since MDF scheme assigns priority to tasks with large data sizes, the computational resources of the UAV cannot be utilized efficiently. Thus, our proposed TDTS algorithm performs better than MDF scheme in terms of the mobility efficiency. Although SDF method has large mobility efficiency, the chosen community may not be optimal to maximize the system throughput since it only considers the affect of distance. Conversely, our proposed TDTS algorithm is able to increase the system throughput at the expense of certain mobility efficiency.

To investigate the effect of the system time period, Fig. 7 exhibits the system throughput under different numbers of time slots. Long system time period enables the UAV to accomplish more tasks. Accordingly, the system throughput increases with the number of time slots in Fig. 7(a). Compared with SDF, MDF and SBS methods, our proposed TDTS algorithm can increase the system throughput by 18%, 17% and 290%, respectively. While in the computational resource-sufficient scenario, the growth trend slows down when the number of time slots exceeds 50000. This is because all tasks are admitted to be uploaded, and can be accomplished before their deadlines. Compared with SDF, MDF and SBS methods, our proposed TDTS algorithm can increase the system throughput by 14%, 9% and 625%, respectively. In addition, SCA method performs almost equally to TDTS algorithm when the number of time slots is relatively small, and the performance gap becomes large with the increase of the system time period. This is because the UAV can complete more tasks during a long time period. These results demonstrate



(a) F = 5 GHz, N = 10 (b) F = 25 GHz, N = 10

Fig. 8. Service percentage with different numbers of time slots.



(a) F = 5 GHz, N = 10 (b) F = 25 GHz, N = 10

Fig. 9. Mobility efficiency with different numbers of time slots.

that the mobility enables the UAV to perform much better than traditional static BS in terms of the system throughput, and our proposed TDTS algorithm can significantly reduce the convergence time at the cost of small performance gap.

We depict the comparison of the service percentage among five algorithms under different numbers of time slots in Fig. 8. Since MDF method prefers to the task with a large data size, few tasks can be accomplished when the number of time slots is small (e.g., $10000 \leq T \leq 30000$). TDTS, SDF and SCA algorithms select tasks with small data sizes, and their performance is similar. When the number of time slots continues to grow, most of tasks with large data size can be accomplished during the system time period. Performance of TDTS and MDF algorithms becomes close. Note that in Fig. 8(b), although the service percentage reaches to 1 when the number of time slots is larger than 30000, the system throughput still increases when the system time period becomes longer. This is because the UAV can select the community where the total data sizes of tasks are larger. Accordingly, the performance of MDF method is superior to that of SDF method, and approximates our proposed TDTS algorithm.

In Fig. 9, the mobility efficiency of four algorithms is compared under different number of time slots. Both in computational resource-limited and resource-sufficient scenarios, SDF method can obtain higher mobility efficiency than the other three algorithms. Our proposed TDTS algorithm performs better than MDF method and close to SCA method, i.e., either the system time period is short (e.g., $T \leq 40000$) or the computational capability of the UAV is poor (e.g., $F = 5$ GHz). Otherwise, tasks with large data sizes can be accomplished before their deadlines, and the gap of the system throughput between TDTS and MDF algorithms narrows. Accordingly, their performance in terms of the mobility efficiency becomes close. Although SDF method has the highest mobility efficiency, it may fail to select the community that can maximize the system throughput due to its shortsightedness. By designing the auction among communities and considering the suffered uplink interferences of users, our proposed TDTS algorithm makes an appropriate trade-off between the mobility distance and the system throughput.

VI. Conclusion

This paper has studied the 5G-enabled UAV-to-community offloading by taking the mobility of UAVs, latency-sensitivity and atomicity of tasks into consideration. Specifically, we have decoupled two decision variables by transforming the system throughput maximization problem into trajectory design and task scheduling subproblems. We have developed an auction mechanism to select one community that can maximize the average throughput. Based on a piecewise function, a community-based latency approximation algorithm has been proposed to calculate the auction bidding. Furthermore, we have designed a dynamic task admission algorithm to maximize the system throughput, while satisfying the latency and atomicity constraints of tasks. Numerical results have demonstrated that an appropriate trade-off between mobility efficiency and system throughput can be made. In both computational resource-insufficient and resource-sufficient scenarios, our proposed TDTS algorithm largely outperforms the other existing solutions.

Since the coverage area of one UAV is limited, multiple UAVs can provide offloading services for a large number of users, where the whole district can be partitioned into several sub-regions according to the local administrative division. Each UAV is responsible for one sub-region and performs our proposed algorithm independently. The trajectory design for the cooperation of multiple UAVs is much complicated than that of single UAV due to the growth of the searching space, and this issue will be investigated in our future work.

References

[1] P. Boccardo, F. Chiabrando, F. Dutto, F. Tonolo, and A. Lingua, "UAV deployment exercise for mapping purposes: Evaluation of emergency response applications," *Sensors*, vol. 15, no. 7, pp. 15717–15737, Jul. 2015.

[2] Z. Ning *et al.*, "Partial computation offloading and adaptive task scheduling for 5G-enabled vehicular networks," *IEEE Trans. Mobile Comput.*, early access, Sep. 18, 2020, doi: 10.1109/TMC.2020.3025116.

[3] X. Wang, Z. Ning, S. Guo, M. Wen, and V. Poor, "Minimizing the age-of-critical-information: An imitation learning-based scheduling approach under partial observations," *IEEE Trans. Mobile Comput.*, early access, Jan. 21, 2021, doi: 10.1109/TMC.2021.3053136.

[4] Z. Ning *et al.*, "Mobile edge computing enabled 5G health monitoring for Internet of medical things: A decentralized game theoretic approach," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 2, pp. 463–478, Feb. 2021.

[5] W. Shi, H. Zhou, J. Li, W. Xu, N. Zhang, and X. Shen, "Drone assisted vehicular networks: Architecture, challenges and opportunities," *IEEE Netw.*, vol. 32, no. 3, pp. 130–137, May 2018.

[6] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, "Dynamic service migration in mobile edge computing based on Markov decision process," *IEEE/ACM Trans. Netw.*, vol. 27, no. 3, pp. 1272–1288, Jun. 2019.

[7] T. Ouyang, Z. Zhou, and X. Chen, "Follow me at the edge: Mobility-aware dynamic service placement for mobile edge computing," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2333–2345, Oct. 2018.

[8] D. Liu, S. Han, C. Yang, and Q. Zhang, "Semi-dynamic user-specific clustering for downlink cloud radio access network," *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 2063–2077, Apr. 2016.

[9] Y. Zeng and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3747–3760, Jun. 2017.

[10] F. Cheng *et al.*, "UAV trajectory optimization for data offloading at the edge of multiple cells," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6732–6736, Jul. 2018.

[11] M. A. Ali, Y. Zeng, and A. Jamalipour, "Software-defined coexisting UAV and WiFi: Delay-oriented traffic offloading and UAV placement," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 6, pp. 988–998, Jun. 2020.

[12] J. Xu, Y. Zeng, and R. Zhang, "UAV-enabled wireless power transfer: Trajectory design and energy optimization," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5092–5106, Aug. 2018.

[13] S. Zhang, H. Zhang, B. Di, and L. Song, "Cellular UAV-to-X communications: Design and optimization for multi-UAV networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 2, pp. 1346–1359, Feb. 2019.

[14] M. Gapeyenko, V. Petrov, D. Moltchanov, S. Andreev, N. Himayat, and Y. Koucheryavy, "Flexible and reliable UAV-assisted backhaul operation in 5G mmWave cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 11, pp. 2486–2496, Nov. 2018.

[15] N. Rupasinghe, Y. Yapici, I. Guvenc, and Y. Kakishima, "Non-orthogonal multiple access for mmWave drone networks with limited feedback," *IEEE Trans. Commun.*, vol. 67, no. 1, pp. 762–777, Jan. 2019.

[16] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-UAV enabled wireless networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 2109–2121, Mar. 2018.

[17] Y. Liu, Y. Li, Y. Niu, and D. Jin, "Joint optimization of path planning and resource allocation in mobile edge computing," *IEEE Trans. Mobile Comput.*, vol. 19, no. 9, pp. 2129–2144, Sep. 2020.

[18] Q. Wu and R. Zhang, "Common throughput maximization in UAV-enabled OFDMA systems with delay consideration," *IEEE Trans. Commun.*, vol. 66, no. 12, pp. 6614–6627, Dec. 2018.

[19] S. Hong *et al.*, "Applications of self-interference cancellation in 5G and beyond," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 114–121, Feb. 2014.

[20] Y. Sun, S. Zhou, and J. Xu, "EMM: Energy-aware mobility management for mobile edge computing in ultra dense networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2637–2646, Nov. 2017.

[21] X. Wang, Z. Ning, and S. Guo, "Multi-agent imitation learning for pervasive edge computing: A decentralized computation offloading algorithm," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 2, pp. 411–425, Feb. 2021.

[22] Z. Ning *et al.*, "Distributed and dynamic service placement in pervasive edge computing networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 6, pp. 1277–1292, Jun. 2021.

[23] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization*. Belmont, MA, USA: Athena Scientific, 1997.

[24] D. Pisinger, *Where are the Hard Knapsack Problems?* Amsterdam, The Netherlands: Elsevier, 2005.

[25] A. J. Kleywegt, A. Shapiro, and T. Homem-de-Mello, "The sample average approximation method for stochastic discrete optimization," *SIAM J. Optim.*, vol. 12, no. 2, pp. 479–502, Jan. 2002.

[26] R. G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. E. Elger, "Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 64, no. 6, Nov. 2001, Art. no. 061907.

[27] Y. Sun, L. Xu, Y. Tang, and W. Zhuang, "Traffic offloading for online video service in vehicular networks: A cooperative approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7630–7642, Aug. 2018.

Zhaolong Ning (Senior Member, IEEE) received the Ph.D. degree from Northeastern University, China, in 2014. He was a Research Fellow with Kyushu University, Japan, from 2013 to 2014. He is currently a Full Professor with the Chongqing University of Posts and Telecommunications. He has published more than 120 scientific articles in international journals and conferences. His research interests include the Internet of Things, mobile edge computing, and resource management. He is a Highly Cited Researcher (Web of Science) and a elected to be the Young Elite Scientists Sponsorship Program by the China Association for Science and Technology. He serves as an Associate Editor or Guest Editor for several journals, such as the IEEE Transactions on Industrial Informatics, IEEE Transactions on Computational Social Systems, and *Computer Journal*.

Peiran Dong (Graduate Student Member, IEEE) received the B.S. and master's degrees from Dalian University of Technology, Dalian, China, in 2018 and 2021, respectively. His research interests include mobile edge computing, network computation offloading, and resource management.

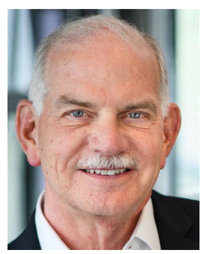
Miaowen Wen (Senior Member, IEEE) received the Ph.D. degree from Peking University, Beijing, China, in 2014.

From 2019 to 2021, he was with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, as a Post-Doctoral Research Fellow. He is currently an Associate Professor with the South China University of Technology, Guangzhou, China. He has published two books and more than 130 journal articles. His research interests include a variety of topics in the areas of wireless and molecular communications. He was a recipient of the IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award in 2020 and four best paper awards from the IEEE ITST'12, the IEEE ITSC'14, the IEEE ICNC'16, and the IEEE ICCT'19. He was also the winner of Data Bakeoff Competition (Molecular MIMO) at the IEEE CTW 2019. He served as a Guest Editor for the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS and the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING. He is currently serving as an Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS, the IEEE TRANSACTIONS ON MOLECULAR, BIOLOGICAL, AND MULTI-SCALE COMMUNICATIONS, and the IEEE COMMUNICATIONS LETTERS.

Xiaojie Wang received the M.S. degree from Northeastern University, China, in 2011, and the Ph.D. degree from the Dalian University of Technology, Dalian, China, in 2019. From 2011 to 2015, she was a Software Engineer with NeuSoft Corporation, China. She is currently a Distinguished Professor at the Chongqing University of Posts and Telecommunications. Her research interests are wireless networks, mobile edge computing, and machine learning.

Lei Guo received the Ph.D. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2006. He is currently a Full Professor with the Chongqing University of Posts and Telecommunications, Chongqing, China. He has authored or coauthored more than 200 technical articles in international journals and conferences. His current research interests include communication networks, optical communications, and wireless communications. He is an editor of several international journals.

Ricky Y. K. Kwok (Fellow, IEEE) received the B.Sc. degree in computer engineering from The University of Hong Kong in 1991 and the M.Phil. and Ph.D. degrees in computer science from The Hong Kong University of Science and Technology (HKUST) in 1994 and 1997, respectively. His research focus has been on designing efficient communication protocols and robust resources management algorithms towards enabling large scale distributed mobile computing. In these research areas, he has authored one textbook, coauthored another two textbooks, and published more than 200 technical articles in various leading journals, research books, and refereed international conference proceedings. He is a fellow of HKIE and IET. From March 2006 to December 2011, he served on the Editorial Board of the *Journal of Parallel and Distributed Computing* as a Subject Area Editor in Peer-to-Peer Computing. He also served as an Associate Editor for the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS from January 2013 to December 2016.



H. Vincent Poor (Life Fellow, IEEE) received the Ph.D. degree in EECS from Princeton University in 1977.

From 1977 to 1990, he was on the faculty of the University of Illinois at Urbana-Champaign. Since 1990, he has been on the faculty at Princeton, where he is currently the Michael Henry Strater University Professor. From 2006 to 2016, he served as the Dean of Princeton's School of Engineering and Applied Science. He has also held visiting appointments at several other universities, including most recently at Berkeley and Cambridge. His research interests are in the areas of information theory, machine learning and network science, and their applications in wireless networks, energy systems, and related fields. Among his publications in these areas is the forthcoming book *Machine Learning and Wireless Communications* (Cambridge University Press, 2021).

Dr. Poor is a member of the National Academy of Engineering and the National Academy of Sciences. He is also a foreign member of the Chinese Academy of Sciences, the Royal Society, and other national and international academies. Recent recognition of his work includes the 2017 IEEE Alexander Graham Bell Medal and a D.Eng. *(honoris causa)* from the University of Waterloo, awarded in 2019.