IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 29, 2020, accepted June 25, 2020, date of publication June 30, 2020, date of current version July 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3006112

Fairness-Aware Offloading and Trajectory Optimization for Multi-UAV Enabled Multi-Access Edge Computing

XIANBANG DIAO[1], MENG WANG[1], JIANCHAO ZHENG[1,2], AND YUEMING CAI[1]

[1]College of Communications Engineering, Army Engineering University of PLA, Nanjing 210007, China
[2]National Innovation Institute of Defense Technology, Academy of Military Sciences of PLA, Beijing 100010, China

Corresponding authors: Meng Wang (teacherdamon@qq.com) and Jianchao Zheng (longxingren.zjc.s@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61801505, in part by the Jiangsu Provincial Nature Science Foundation of China under Grant BK20170755, and in part by the Young Elite Scientist Sponsorship Program.

ABSTRACT Multiple unmanned aerial vehicles (UAVs) can compensate for the performance deficiencies of a single UAV in multi-access edge computing (MEC) systems, thus providing improved offloading services to user equipments (UEs). In multi-UAV enabled MEC systems, the offloading strategy and UAVs' trajectories affect the fairness of both UEs and UAVs, which affects the UE experience and UAVs' existence durations. Therefore, we investigate fairness-aware offloading and trajectory optimization in the system. To ensure fairness of energy consumptions (ECs) for both UEs and UAVs, we minimize the weighted sum of the maximum EC among UEs and the maximum EC among UAVs subject to the task delay, the offloading strategy and UAVs' trajectories constraints. Despite the non-convexity of the original formulated joint optimization problem, we transform the problem into two sub-problems and solve them one by one. Finally, an iterative optimization algorithm is proposed to alternately optimize the offloading strategies and the UAVs' trajectories. Simulation results show that the proposed algorithm can effectively reduce both the maximum EC among UEs and the maximum EC among UAVs and ensure the fairness of both the UEs and UAVs.

INDEX TERMS Fairness, multi-access edge computing, multiple UAVs, trajectory optimization, offloading optimization.

I. INTRODUCTION

In recent years, smart mobile applications (e.g., speech recognition, augmented reality, sensing data processing) [1]–[4] have brought convenience to people's lives, but also brought great challenges to user equipments (UEs). These smart applications are computation-intensive and resource-hungry applications that require large amounts of computing resources and energy consumptions (ECs). Due to resources constraints, the UEs can hardly bear the EC caused by executing these applications locally, which prevents the smart applications from being applied on the UEs [5], [6].

Multi-Access edge computing (MEC) [7]–[9], deploying computing servers close to the UEs, has been considered as an effective technology to address this problem. In traditional MEC, the computing servers are generally fixed in the base stations (BSs) or access points (APs). This service method mainly has the following disadvantages. Firstly, it is difficult for fixed servers to cope with emergency computing needs in disaster areas. Moreover, it is difficult for fixed servers to be deployed in the areas where there are computing requirements but the BS/AP is difficult to set up. In addition, fixed servers can not satisfy the computing requirements of remote UEs [10]. Therefore, a flexible computing service method should be studied.

Recently, unmanned aerial vehicles (UAVs) attract a lot of attentions in wireless communications due to the flexibility and maneuverability [11], [12]. According to the existing studies [13]–[17], the UAVs can be flexibly and quickly deployed over the area of interest to provide emergency and considerate services to ground UEs. Thus, the UAVs installed with computing servers can provide computing services for all UEs in a location-flexible way. Moreover, since the channel between UAVs and UEs can be modeled as

The associate editor coordinating the review of this manuscript and approving it for publication was Huaqing Li.

line-of-sight (LoS) links [18], the transmission rate of UEs can be improved. In general, since the UAV-enabled MEC system has many advantages, it has broad application prospects.

A. RELATED WORK AND MOTIVATION

There are some literatures that have studied UAV-enabled MEC systems. For example, in [19], Zhou *et al.* regarded the UAV as an aerial platform which not only transfers energy to UEs, but also provides UEs with computing services. In [20], Hu *et al.* jointly optimized the UAV's trajectory, the ratio of offloading tasks, and the user scheduling strategy to minimize the sum of the maximum delay among UEs in each time slot. In addition, there are some studies investigating the EC optimization for UAV-enabled MEC systems. In [21], Jeong *et al.* minimized the sum EC of all UEs by jointly optimizing trajectory, task data and power allocations. In [22], Hu *et al.* minimized the sum EC of all UEs by optimizing the UAV's hover position, and the duration and ratio of offloading tasks. In [23], Hua *et al.* minimized the total EC of UEs with one-by-one access scheme. Moreover, some works studied the optimization of the EC of UAVs or the weighted EC sum of the UEs and the UAV. In [24], Zhang *et al.* considered the task queues of the UEs and the UAV, and minimized the weighted EC sum of the UEs and UAV by optimizing resources allocations and UAV's trajectory. In [25], Hu *et al.* studied the UAV-assisted relaying and edge computing system and minimized weighted EC sum of UEs and UAV by optimizing the computing rates, bandwidth allocation and UAV's trajectory. In [26], Du *et al.* studied the UAV-enabled wireless-powered MEC system and minimized the UAV's EC by optimizing the offloading strategies, computing resource allocation, and the hovering duration and wireless charging duration of the UAV. In [27], Liu *et al.* investigated the UAV-enabled wireless powered cooperative MEC system and minimized the UAV's EC by optimizing the computing rates, the amount of offloading task and the UAV's transmit power and trajectory.

The above studies focus on the simplified case of single UAV. However, as a matter of fact, the real computing platform usually consists of multiple UAVs, since a single UAV can not provide high quality computing services to all UEs for a long time due to physical limitations, such as computing resources and battery capabilities. There are only a few literatures studying multi-UAV enabled MEC systems. In [28], Yang *et al.* studied the multi-UAV enabled MEC system and minimized the weighted EC sum of all UEs and UAVs in the system by optimizing offloading strategies, UAVs' hovering positions and computing rates. In [29], Wang *et al.* optimized the number and hovering positions of UAVs, offloading strategies and computing resource allocation to minimize the weighted EC sum of UEs and UAVs in a static multi-UAV enabled MEC system. In [30], Wang *et al.* maximized number of UEs whose task can be completed within the maximal tolerance time under predetermined maximum number of UAVs and an online multiple UAV-enabled edge server dispatching scheme was proposed to provide flexible computing services. In [31], Yang *et al.* optimized the deployment of UAVs and task scheduling to minimize task delays and balance the UAVs' loads. The above works both studied static multi-UAV scenarios, however, keeping the UAVs hovering does not take advantage of UAVs' flexibility and can not further explore the performance improvements brought by multiple UAVs. Therefore, it motivated us to study the dynamic multi-UAV enabled MEC system.

B. CONTRIBUTIONS AND ORGANIZATION

In this paper, we study a multi-UAV enabled MEC system where multiple UAVs roaming in the area of interest help UEs complete the computing tasks. Reaping the benefit of multiple UAVs for MEC systems requires to take into account the following key issues. Firstly, the undesigned offloading strategies and UAVs' trajectories may cause large EC gaps between UEs, which affect the service experience of some UEs. Second, to serve all UEs fairly, energy-limited UAVs tend to be closer to each UE, resulting in non-negligible ECs, thus reducing the existence duration of UAVs. Thus, the EC of UAVs should also be an indicator of the system performance. Third, multiple UAVs have trajectory differences when serving UEs, thus causing EC fairness issues. If a UAV can not continue to work due to excessive flight EC, it will seriously affect the performance and even lead to the failure of the multi-UAV system. As a result, it is of practical importance for us to take into account the EC fairness of both UEs and UAVs.

Therefore, we jointly optimize the offloading strategies and UAVs' trajectories to minimize the weighted sum of the maximum EC among the UEs and the maximum EC among the UAVs. Although the integer offloading strategy variables and the ECs of UAVs in the non-convex problem are difficult to be addressed, we propose corresponding methods to solve these problems. Specifically, we propose a greedy-based offloading strategy rounding algorithm which utilizes greedy rules to make the performance of the reconstructed integer offloading strategies close to that of the optimal linear offloading strategies. Moreover, a upper bound method is used to convert the non-convex UAVs' ECs into convex upper bounds of ECs. Denote that there is a central controller (e.g., a large rotor UAV at a higher altitude), which collects UEs' computing request packet which includes positions and computing task information, optimizes computing offloading strategies and UAVs' trajectories, and finally releases the offloading strategies and trajectories to UEs and UAVs, respectively. It is worth noting that in our system, the data amounts of the computing request packets sent by UEs and the optimized results returned by the controller are small. Therefore, compared to the total task time, the transmission delays of collecting UEs' information and returning the optimized results are small and can be ignored. In addition, the computing ability of the controller is generally strong, so the processing delay of the central controller is also ignored. The main contributions of the paper are summarized as follows.

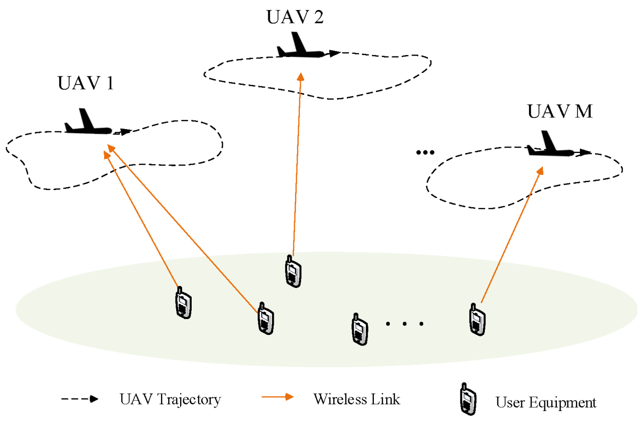


FIGURE 1. **The multi-UAV enabled MEC system.**

1) We consider a fairness-aware multi-UAV enabled MEC system in which the UAVs can make full use of their flexibility to provide UEs with fair computing services and the UAVs also ensure EC fairness with other UAVs when serving UEs. The system can extend the service duration of multiple UAVs while ensuring the EC fairness among UEs.
2) The Min-Max fairness is considered as a measure of ensuring the fairness of both UEs and UAVs and the fairness problem is model as a weighted multi-objective optimization problem. Specifically, we minimize the weighted sum of the maximum EC among the UEs and the maximum EC among the UAVs.
3) The original non-convex problem is divided into two sub-problems. For the offloading strategies, we convert the integer problem to a linear problem and then propose a greedy-based rounding algorithm to obtain a near-optimal integer solution. For the UAVs' trajectories, we obtain the optimal solution for the upper bound problem. Finally, we propose an iterative algorithm to alternately optimize the offloading strategies and trajectories to obtain a sub-optimal solution.

The rest of this paper is organized as follows. In Section II, we introduce the system model and formulate the optimization problem. In Section III, we propose an iterative algorithm to solve the optimization problem. Section IV presents the numerical results. Finally, we conclude the paper in Section V.

II. SYSTEM MODEL AND PROBLEM FORMULATION

As shown in Fig. 1, we consider a multi-UAV enabled MEC system where M fixed-wing UAVs with computing abilities roaming in the area of interest provide computing services for K UEs, denoted by $\mathcal{K} = \{1, 2, \ldots, K\}$. In a finite time horizon T, we assume that all UEs' locations remain unchanged, and each UE k has a task which adopts the partial computation offloading mode, where the task data is bitwise independent and can be arbitrarily divided into different groups [7]. To make the flight more trackable, T is divided into N equal time slots τ, i.e., $T = N\tau$, and denote $\mathcal{N} = \{1, 2, \ldots, N\}$. When τ is sufficiently small, the position of the UAV during each τ can be regarded as stationary. In the n-th slot, each UE k can calculate the sub-task $A_{k,n} \triangleq (D_{k,n}, X_{k,n})$ locally or offload it to one UAV with the frequency division multiple access (FDMA) protocol. $D_{k,n}$ is the input-data size of the UE k in the n-th time slot (in bits), and $D_{k,n} = \frac{D_k}{N}, \forall k \in \mathcal{K}, n \in \mathcal{N}$, where D_k is total amount of task data of UE k. $X_{k,n}$ represents the computing intensity assigned to the UE k in the n-th time slot (in the unit of CPU cycles per bit). Moreover, due to limitations of UAVs' computing abilities, we assume that each UAV only serve up to U_{max} UEs. When the UE k chooses to offload its sub-task to one UAV, its bandwidth is $B = \frac{B_{\text{total}}}{MU_{\max}}$, where B_{total} is the total bandwidth. We construct a 3-Dimensional Cartesian coordinate system model, where the coordinate of the UE k is $\mathbf{u}_k = [x_k, y_k, 0]^T, \forall k \in \mathcal{K}$ and the coordinate of the UAV m in the n-th time slot is $\mathbf{q}_m[n] = [x_m[n], y_m[n], H]^T, \forall n \in \mathcal{N}$, where the H is a constant. Moreover, we define the set of all UAVs' trajectories as $\mathbf{Q} = \{\mathbf{q}_1[n], \ldots, \mathbf{q}_M[n], \forall n \in \mathcal{N}\}$.

In the n-th time slot, UEs can choose local computing or computing offloading. When the UE k selects local computing in the n-th time slot, the task delay and the EC of the UE k can be respectively calculated as

$$T_k^{\text{local}}[n] = \frac{X_{k,n}D_{k,n}}{f_{k,n}}, \tag{1a}$$

$$E_k^{\text{local}}[n] = \kappa X_{k,n}D_{k,n}f_{k,n}^2, \tag{1b}$$

where κ is a coefficient depending on the chip architecture [32]. The $f_{k,n}$ is the local computing rate of the UE k in the n-th time slot (in the unit of CPU cycles per sencond). Moreover, the maximum computing rate of the UE k is f_k^{max}.

Similar to [14]–[16], the channels between UAVs and UEs are dominated by line-of-sight links. Therefore, the channel gain between the UAV m and the UE k in the n-th slot is given by

$$g_{k,m}[n] = \frac{g_0}{\|\mathbf{q}_m[n] - \mathbf{u}_k\|^2}, \tag{2}$$

where parameter g_0 represents the channel power gain at the reference distance $d_0 = 1$ m, and the $\|\cdot\|$ represents the norm operator. Moreover, the transmission rate from the UE k to the UAV m in the n-th slot is given by

$$R_{k,m}[n] = B\log_2\left(1 + \frac{p_k g_{k,m}[n]}{N_0 B}\right), \tag{3}$$

where p_k is the transmission power of the UE k and N_0 is the noise power density. When the UE k chooses to offload the sub-task $A_{k,n}$ to the UAV m, the task delay and EC of the UE k are are respectively given by

$$T_{k,m}^{\text{off}}[n] = \frac{D_{k,n}}{R_{k,m}[n]} + \frac{X_{k,n}D_{k,n}}{F}, \tag{4a}$$

$$E_{k,m}^{\text{off}}[n] = \frac{p_k D_{k,n}}{R_{k,m}[n]}, \tag{4b}$$

where F represents the computing rate assigned by the UAV m to one of the UEs accessing it, and is a constant. In other words, the total computing ability of an UAV is $U_{max}F$. In (4a), similar to [5], we do not consider the delay for sending back the computing results. Moreover, we define a set $\mathcal{M} = \{0, 1, 2, \ldots, M\}$ and the binary variable $s_{k,m}[n]$, which represents the computing strategy of the UE k in the n-th time slot. When $s_{k,m}[n] = 1, \forall m \in \mathcal{M}\backslash\{0\}$, the UE k offloads the sub-task $A_{k,n}$ to the UAV m in n-th slot. Particularly, $s_{k,0}[n] = 1$ indicates the UE k performs local computing in the n-th time slot. Moreover, we define $\mathbf{S} = \{s_{k,m}[n], \forall k \in \mathcal{K}, m \in \mathcal{M}, n \in \mathcal{N}\}$. In summary, the task delay and EC of the UE k in the n-th time slot are given by

$$T_k[n] = s_{k,0}[n]T_k^{\text{local}}[n] + \sum_{m=1}^{M} s_{k,m}[n]T_k^{\text{off}}[n], \tag{5a}$$

$$E_k[n] = s_{k,0}[n]E_k^{\text{local}}[n] + \sum_{m=1}^{M} s_{k,m}[n]E_k^{\text{off}}[n]. \tag{5b}$$

In the T period, UAVs constantly fly and calculate tasks for UEs. Therefore, the total EC of an UAV includes computing and flight ECs. Moreover, since the τ is small enough, the UAVs can be considered to fly with a constant speed in a slot. Therefore, similar to [25], the total EC of UAV m in the n-th slot can be expressed as

$$E_m[n] = \sum_{k=1}^{K} s_{k,m}[n]\kappa X_{k,n}D_{k,n}F^2 + \tau\left(k_1\left\|\mathbf{v}_m[n]\right\|^3 + \frac{k_2}{\left\|\mathbf{v}_m[n]\right\|}\right), \tag{6}$$

where $\|\mathbf{v}_m[n]\| = \frac{\|\mathbf{q}_m[n+1]-\mathbf{q}_m[n]\|}{\tau}$ represents the velocity of the UAV m in the n-th slot. k_1 and k_2 are two parameters related to the UAV's weight, wing area, air density, etc.

In order to ensure the Min-Max fairness [33] of UEs while considering the EC fairness of multiple UAVs, we aim to minimize the weighted sum of the maximal ECs among all UEs and the maximal ECs among all UAVs by jointly optimizing the computing offloading strategies and UAVs' trajectories. The optimization problem can be formulated as follows

$$\mathbf{P1}: \min_{\mathbf{Q},\mathbf{S}} \omega_k \max_k\{\sum_{n=1}^{N} E_k[n]\} + \omega_m \max_m\{\sum_{n=1}^{N} E_m[n]\}, \tag{7a}$$

$$\text{s.t. } T_k[n] \le \tau, \quad \forall k \in \mathcal{K},\ n \in \mathcal{N}, \tag{7b}$$

$$\mathbf{q}_m[1] = \mathbf{q}_m[N+1] = \mathbf{q}_m, \quad \forall m \in \mathcal{M}, \tag{7c}$$

$$\|\mathbf{q}_m[n+1] - \mathbf{q}_m[n]\| \le \tau V_{\max}, \quad \forall n \in \mathcal{N}, \tag{7d}$$

$$\|\mathbf{q}_m[n+1] - \mathbf{q}_m[n]\| \ge \tau V_{\min}, \quad \forall n \in \mathcal{N}, \tag{7e}$$

$$\left\|\mathbf{q}_x[n] - \mathbf{q}_y[n]\right\| \ge d_{\min}, \quad \forall n \in \mathcal{N}, \quad \forall x, y \in \mathcal{M},\ x \ne y, \tag{7f}$$

$$s_{k,m}[n] \in \{0, 1\}, \quad \forall k \in \mathcal{K},\ m \in \mathcal{M},\ n \in \mathcal{N}, \tag{7g}$$

$$\sum_{k=1}^{K} s_{k,m}[n] \le U_{\max}, \quad \forall m \in \mathcal{M},\ n \in \mathcal{N}, \tag{7h}$$

$$\sum_{m=0}^{M} s_{k,m}[n] = 1, \quad \forall k \in \mathcal{K}, \tag{7i}$$

where ω_k and ω_m are positive weighted factors for UEs and UAVs, respectively. Their effect is to reduce the EC gap between UEs and UAVs so that their ECs can be optimized better. Here, constraint (7b) represents the task delay of any UE in any time slot must be less than the time slot length τ. Constraint (7c) represents that the initial and final points of any UAV's trajectory are same and fixed. Constraints (7d) and (7e) indicate that the fixed-wing UAV's flight rate must be less than the maximal reachable velocity V_{max}, and must be greater than the minimum velocity V_{min} to remain aloft. Constraint (7f) represents the distance between any two UAVs can not be less than the minimal safety distance d_{min} in any time slot. Constraint (7g) indicates that the offloading strategy variables are binary variables. Constraint (7h) represents that any UAV serves up to U_{max} UEs. Constraint (7i) ensures any UE must choose only one computing method at any time slot. Since the objective function and the constraints (7b) and (7e)-(7g) are non-convex, the problem is a non-convex optimization problem and can not be directly solved by the convex optimization techniques.

III. PROPOSED ALGORITHM

In this section, we transform the problem **P1** into two sub-problems and solve them one by one. Then, we propose an algorithm to alternately solve the offloading strategies and UAVs' trajectories. Specifically, since any UE's strategy is either local computing or computing offloading, we convert the constraint (7b) into the following form

$$\sum_{m=1}^{M} s_{k,m}[n]\left(\frac{D_{k,n}}{R_{k,m}[n]} + \frac{X_{k,n}D_{k,n}}{F}\right) \le \tau, \quad \forall k \in \mathcal{K},\ n \in \mathcal{N}, \tag{8a}$$

$$f_{k,n} \ge \frac{s_{k,0}[n]X_{k,n}D_{k,n}}{\tau}, \quad \forall k \in \mathcal{K},\ n \in \mathcal{N}. \tag{8b}$$

In order to make $f_{k,n}$ satisfy the constraint (8b), we assume $f_k^{\max} \ge \frac{X_{k,n}D_{k,n}}{\tau}$. Based on (1b), in order to reduce local ECs of UEs, we set $f_{k,n} = \frac{X_{k,n}D_{k,n}}{\tau}$. Moreover, we introduce the auxiliary variables $e_1 \ge \max_k\left\{\sum_{n=1}^{N} E_k[n]\right\}$ and $e_2 \ge \max_m\left\{\sum_{n=1}^{N} E_m[n]\right\}$ to simplify the objective function. Thus, the problem **P1** can be equivalently reformulated as follows

$$\mathbf{P2}: \min_{\mathbf{Q},\mathbf{S},e_1,e_2} \omega_k e_1 + \omega_m e_2, \tag{9a}$$

$$\text{s.t. } e_1 \ge \sum_{n=1}^{N}\left(s_{k,0}[n]\kappa X_{k,n}D_{k,n}f_{k,n}^2 + \sum_{m=1}^{M}\frac{s_{k,m}[n]p_kD_{k,n}}{R_{k,m}[n]}\right), \quad \forall k \in \mathcal{K}, \tag{9b}$$

$$e_2 \geq \sum_{n=1}^{N} \left(\begin{array}{l} \sum\limits_{k=1}^{K} s_{k,m}[n]\kappa X_{k,n} D_{k,n} F^2 \\ +\tau \left(k_1 \left\| \mathbf{v}_m[n] \right\|^3 + \frac{k_2}{\left\| \mathbf{v}_m[n] \right\|} \right) \end{array} \right), \quad \forall m \in \mathcal{M}, \tag{9c}$$

$$\text{Constraints (7c)-(7i),(8a).} \tag{9d}$$

In order to solve the non-convex problem **P2**, we solve two sub-problems under given variables, and then jointly solve all sub-problems.

A. OFFLOADING STRATEGY OPTIMIZATION

For problem **P2** with fixed **Q**, the offloading strategy optimization problem can be formulated as

$$\mathbf{P2.1}: \min_{\mathbf{S}, e_1} \omega_k e_1 + \omega_m e_2, \tag{10a}$$

$$\text{s.t.} \quad \text{Constraints (7g)-(7i),(8a),(9b)-(9c),} \tag{10b}$$

Since the **S** is a set of binary variables, the problem **P2.1** is a non-convex problem, and it is difficult to find the optimal solution by using exhaustive search algorithms. The branch-and-bound method [34] is an alternative method, however, which is time-consuming because the **S** contains a large number of variables. To solve the NP-hard problem, we first relax the $s_{k,m}[n]$ to continuous variables with a numerical interval [0,1], namely, $0 \leq s_{k,m}[n] \leq 1, \forall k \in \mathcal{K}, m \in \mathcal{M}, n \in \mathcal{N}$. After the above operation, the problem **P2.1** is transformed into a linear programming problem **P2.1.1**. It is worth noting that the ECs of the UEs are much smaller than the ECs of the UAVs. In order to enable the UEs' ECs and UAVs' ECs to be optimized better. ω_k should be much larger than ω_m, which makes the small computing ECs of UAVs account for a very small proportion of the weighted EC sum. Therefore, the problem **P2.1.1** can be simplified as minimizing the maximal EC among UEs, i.e., the optimization goal of the problem **P2.1.1** can be simplified as $\omega_k e_1$. Moreover, we utilize CVX [35] to solve the problem and find the linear optimal solution.

Algorithm 1 GOSVR-Algorithm: To Reconstruct Offloading Strategy Variables

```
Input: $K$, $M$, $N$, and optimal continuous variables $\{sc_{k,m}[n]\}$ obtained by solving the problem P2.1.1.
Set $m = 1, n = 1, \{s_{k,m}[n]\} = 0$.
repeat
    repeat
        According to the descending order of $\{sc_{k,m}[n], \forall k \in \mathcal{K}\}$, obtain the corresponding rank of the UEs' number $\mathbf{rankue}_{K\times 1}$.
        $\mathbf{RankUE}(1:K, m, n) = \mathbf{rankue}$, $n = n + 1$.
    until $n > N$
    $m = m + 1, n = 1$.
until $m > M$
Set $n = 1$.
repeat
    Set arrays $\mathbf{RSC}_{1\times M}$ and $\mathbf{flag}_{1\times K}$ whose elements are both equal to $U_{max}$ and 0, respectively, and use an array set $\mathbf{UAV} = \{UAV1_{K\times 1}, \ldots, UAVM_{K\times 1}\}$ to accommodate all M columns of $\mathbf{RankUE}(1:K, 1:M, n)$, and set $m = 1$.
    repeat
        Set $k = 1$.
        repeat
            if $\mathbf{flag}(k) == 1$ then
                $k = k + 1$.
                Continue.
            end if
            According to the descending order of $\{sc_{k,m}[n], \forall m \in \mathcal{M}\}$, obtain the corresponding rank of the computing body number $\mathbf{rankcb}_{1\times M+1}$.
            $m^* = \mathbf{rankcb}(m)$.
            if $m^* = 0$ then
                $s_{k,0}[n] = 1$, $\mathbf{flag}(k) = 1$, and delete the UE number $k$ in all arrays of UAV.
            else if $UAVm^*(1:\mathbf{RSC}(m^*)) \cap k == k$ then
                $s_{k,m^*}[n] = 1$, $\mathbf{RSC}(m^*) = \mathbf{RSC}(m^*) - 1$, $\mathbf{flag}(k) = 1$, and delete the UE number $k$ in all arrays in UAV.
            end if
            $k = k + 1$.
        until $k > K$
        $m = m + 1$.
    until $m > M + 1$
    $n = n + 1$.
until $n > N$
Output: $\{s_{k,m}[n]\}$.
```

However, $\{s_{k,m}[n]\}$ obtained by the above method is a series of decimals. Therefore, we propose a greedy-based offloading strategy variable rounding (GOSVR) algorithm to reconstruct $\{s_{k,m}[n]\}$. In this algorithm, we regard $s_{k,m}[n]$ as the UE k's preference value for the computing body m, where the computing body is UE k when $m = 0$, and the computing body is UAV m when $m \neq 0$. We assume that each UE is greedy, namely, the larger $s_{k,m}[n]$ is, the more the UE k tends to perform the sub-task on the computing body m in n-th slot. Therefore, under constraints (7h), we select the computing body whose preference value is as large as possible for each UE in each slot. Specifically, the algorithm obeys the following principles

1) **Greedy principle**: In each slot, each UE tends to select a computing body with the largest preference value. If the computing body can not be selected by the UE, the UE will observe whether the computing body with the second largest preference value can be selected, and so on, until find an alternative computing body.
2) **Selection principle**: If the UE wants to select itself as the computing body, it can directly select itself. If the UE wants to select UAV m as its computing body,

its preference value for the UAV m must be in the top $\mathbf{RSC}(m)$ among all UEs that have not determined their computing bodies, where $\mathbf{RSC}(m)$ represents the remaining service capacity of the UAV m.

3) **Constraint principle**: If $\mathbf{RSC}(m) = 0$, the remaining UEs that have not determined their computing bodies can not choose the UAV m.

The GOSVR-Algorithm is illustrated in Algorithm 1. Moreover, in the algorithm, the *UAVm* represents the m-th array of the array set **UAV**. In general, the GOSVR-Algorithm can make the performance of the reconstructed integer offloading strategies approach the performance of the optimal linear offloading strategies from two aspects, so as to obtain an near-optimal solution. On the one hand, the problem **P2.1.1** is a minimization problem of EC. For any UE, the EC of the computing body m with the largest $sc_{k,m}[n]$ is the smallest, where the $sc_{k,m}[n]$ belongs to the optimal linear offloading strategies. Therefore, the GOSVR-Algorithm allows the UE to greedily select the optional computing body m with the larger $sc_{k,m}[n]$, which meets the requirement of EC minimization. On the other hand, since the problem **P2.1.1** considers UEs' EC fairness, the allocation of $sc_{k,m}[n]$ can make the UEs' ECs tend to be fair. Specifically, the fairness is reflected when UEs occupy the limited computing resources of UAVs. In a specific time slot n, a larger $sc_{k,m}[n]$ represents that the UE k occupies more computing resources of the UAV m. During the entire time period T, all UEs occupy the computing resources of UAVs fairly with the change of the time slots, namely, the UEs occupy more computing resources of UAVs in turn. Therefore, the GOSVR-Algorithm selects the UEs with larger $sc_{k,m}[n]$ as the service targets of the UAV m, which meets the requirements of fairness.

B. TRAJECTORY OPTIMIZATION

For problem **P2** with fixed **S**, the trajectory optimization problem can be formulated as

$$\mathbf{P2.2}: \min_{\mathbf{Q},e_1,e_2} \ \omega_k e_1 + \omega_m e_2, \tag{11a}$$

$$\text{s.t. Constraints (7c)-(7f),(8a),(9b)-(9c).} \tag{11b}$$

Since constraints (7c)-(7f), (8a), (9b)-(9c) are non-convex, problem **P2.2** is a non-convex problem. Therefore, we introduce auxiliary variables to convert it into a convex problem. First, we introduce auxiliary variables $\{t_{k,m}[n]\}$, which satisfy the following constraints

$$\frac{1}{t_{k,m}[n]} \leq R_{k,m}[n], \quad \forall k \in \mathcal{K},\ m \in \mathcal{M},\ n \in \mathcal{N}, \tag{12a}$$

$$t_{k,m}[n] > 0, \quad \forall k \in \mathcal{K},\ m \in \mathcal{M},\ n \in \mathcal{N}, \tag{12b}$$

where $R_{k,m}[n]$ is not a concave function with respect to $\mathbf{q}_m[n]$, but $R_{k,m}[n]$ is a convex function with respect to $\|\mathbf{q}_m[n] - \mathbf{u}_k\|^2$. Therefore, for any given feasible local point $\left\|\mathbf{q}_m^{\text{local}}[n] - \mathbf{u}_k\right\|^2$, the following inequality holds

$$R_{k,m}^{\text{low}}[n] = \frac{B}{\ln 2} \ln\left(1 + \frac{\alpha}{\left\|\mathbf{q}_m^{\text{local}}[n] - \mathbf{u}_k\right\|^2}\right) - \frac{B\alpha\left(\|\mathbf{q}_m[n] - \mathbf{u}_k\|^2 - \left\|\mathbf{q}_m^{\text{local}}[n] - \mathbf{u}_k\right\|^2\right)}{\ln 2\left(\left\|\mathbf{q}_m^{\text{local}}[n] - \mathbf{u}_k\right\|^4 + \alpha\left\|\mathbf{q}_m^{\text{local}}[n] - \mathbf{u}_k\right\|^2\right)} \leq R_{k,m}[n], \quad \forall k \in \mathcal{K},\ m \in \mathcal{M},\ n \in \mathcal{N}, \tag{13}$$

where $\alpha = \frac{p_k g_0}{N_0 B}$, and $R_{k,m}^{\text{low}}[n]$ is a concave function with respect to $\mathbf{q}_m[n]$. Therefore, the constraint (12a) can be written as

$$\frac{1}{t_{k,m}[n]} \leq R_{k,m}^{\text{low}}[n], \quad \forall k \in \mathcal{K},\ m \in \mathcal{M},\ n \in \mathcal{N}. \tag{14}$$

Since the structures of constraints (7e) and (7f) are similar, we take the processing method of (7e) as an example. Due to $\|\mathbf{q}_m[n+1] - \mathbf{q}_m[n]\| > 0$ and $\tau V_{\min} > 0$, the constraint (7e) can be written as $\|\mathbf{q}_m[n+1] - \mathbf{q}_m[n]\|^2 \geq \tau^2\ V_{\min}^2$. Since $\|\mathbf{q}_m[n+1] - \mathbf{q}_m[n]\|^2$ is a convex function with respect to $\|\mathbf{q}_m[n+1] - \mathbf{q}_m[n]\|$, under any given feasible local point $\left\|\mathbf{q}_{m,n}^{\text{local}}\right\| = \left\|\mathbf{q}_m^{\text{local}}[n+1] - \mathbf{q}_m^{\text{local}}[n]\right\|$, the constraint (7e) can be rewritten as

$$\left\|\mathbf{q}_{m,n}^{\text{local}}\right\|^2 + 2\left(\mathbf{q}_{m,n}^{\text{local}}\right)^T\left(\mathbf{q}_m[n+1] - \mathbf{q}_m[n] - \mathbf{q}_{m,n}^{\text{local}}\right) \geq \tau^2 V_{\min}^2, \quad \forall m \in \mathcal{M},\ n \in \mathcal{N}. \tag{15}$$

Similarly, under any given feasible local $\left\|\mathbf{q}_{x,y}^{\text{local}}\right\| = \left\|\mathbf{q}_x^{\text{local}}[n] - \mathbf{q}_y^{\text{local}}[n]\right\|$, the constraint (7f) can be rewritten as

$$\left\|\mathbf{q}_{x,y}^{\text{local}}\right\|^2 + 2\left(\mathbf{q}_{x,y}^{\text{local}}\right)^T\left(\mathbf{q}_x[n] - \mathbf{q}_y[n] - \mathbf{q}_{x,y}^{\text{local}}\right) \geq d_{\min}^2, \quad \forall n \in \mathcal{N}, \quad \forall x, y \in \mathcal{M},\ x \neq y. \tag{16}$$

Moreover, the constraint (9c) is not convex with respect to $\mathbf{v}_m[n]$, we propose a UAV's EC upper bound method to convert it to a convex constraint. Specifically, since we previously assume that the UAVs fly with constant velocities in each time slot, the EC of each UAV can be written as the product of the resistance and distance. Therefore, the constraint (9c) can be rewritten as follows

$$e_2 \geq \sum_{n=1}^{N}\left(\begin{array}{l}\sum_{k=1}^{K} s_{k,m}[n]\kappa X_{k,n} D_{k,n} F^2 + \\ Drag(\|\mathbf{v}_m[n]\|)\,\|\mathbf{q}_m[n+1] - \mathbf{q}_m[n]\|\end{array}\right), \quad \forall m \in \mathcal{M}, \tag{17}$$

where $Drag(\|\mathbf{v}_m[n]\|) = k_1\|\mathbf{v}_m[n]\|^2 + \frac{k_2}{\|\mathbf{v}_m[n]\|^2}$. Since the second derivative of $Drag(\|\mathbf{v}_m[n]\|)$ is $2k_1 + \frac{6k_2}{\|\mathbf{v}_m[n]\|^4}$, the $Drag(\|\mathbf{v}_m[n]\|)$ is convex function with respect to $\|\mathbf{v}_m[n]\|$. Therefore, the maximal value of *Drag* in the numerical interval $[V_{min}, V_{max}]$ can be obtained by substituting the end values of the interval, namely, $Drag_{\max} = \max\{Drag(V_{\min}), Drag(V_{\max})\}$. Thus, constraint (17) can be written as

$$e_2 \geq \sum_{n=1}^{N}\left(\begin{array}{l} s_{k,m}[n]\kappa X_{k,n} D_{k,n} F^2 + \\ Drag_{\max}\,\|\mathbf{q}_m[n+1] - \mathbf{q}_m[n]\|\end{array}\right), \quad \forall m \in \mathcal{M}. \tag{18}$$

In summary, based on the above transformation and processing, the problem **P2.2** can be formulated as

$$\begin{aligned}
\mathbf{P2.2.1}:\ &\min_{\mathbf{Q},t_{k,m}[n],e_1,e_2} \quad w_k e_1 + w_m e_2, && \text{(19a)}\\
&\text{s.t.} e_1 \geq \sum_{n=1}^{N}\left(\lambda_{k,n} + \sum_{m=1}^{M}\mu_{k,m,n}t_{k,m}[n]\right),\\
&\qquad \forall k\in\mathcal{K}, n\in\mathcal{N}, && \text{(19b)}\\
&e_2 \geq \sum_{n=1}^{N}\left(\begin{array}{l} s_{k,m}[n]\kappa X_{k,n}D_{k,n}F^2 + \\ Drag_{\max}\left\|\mathbf{q}_m[n+1]-\mathbf{q}_m[n]\right\| \end{array}\right),\\
&\qquad \forall m\in\mathcal{M}, && \text{(19c)}\\
&\sum_{m=1}^{M} s_{k,m}[n]\left(D_{k,n}t_{k,m}[n] + \frac{X_{k,n}D_{k,n}}{F}\right) \leq \tau,\\
&\qquad \forall k\in\mathcal{K},\ n\in\mathcal{N}, && \text{(19d)}\\
&\mathbf{q}_m[1] = \mathbf{q}_m[N+1] = \mathbf{q}_m, \quad \forall m\in\mathcal{M}, && \text{(19e)}\\
&\left\|\mathbf{q}_m[n+1]-\mathbf{q}_m[n]\right\| \leq \tau V_{\max}, \quad \forall n\in\mathcal{N}, && \text{(19f)}\\
&\left\|\mathbf{q}_{m,n}^{\text{local}}\right\|^2 + 2\left(\mathbf{q}_{m,n}^{\text{local}}\right)^T\\
&\quad\times\left(\mathbf{q}_m[n+1]-\mathbf{q}_m[n]-\mathbf{q}_{m,n}^{\text{local}}\right)\\
\geq\ &\tau^2 V_{\min}^2, \quad \forall m\in\mathcal{M},\ n\in\mathcal{N}, && \text{(19g)}\\
&\left\|\mathbf{q}_{x,y}^{\text{local}}\right\|^2 + 2\left(\mathbf{q}_{x,y}^{\text{local}}\right)^T\\
&\quad\times\left(\mathbf{q}_x[n]-\mathbf{q}_y[n]-\mathbf{q}_{x,y}^{\text{local}}\right) \geq d_{\min}^2,\\
&\forall n\in\mathcal{N},\quad \forall x,y\in\mathcal{M},\ x\neq y, && \text{(19h)}\\
&\frac{1}{t_{k,m}[n]} \leq R_{k,m}^{\text{low}}[n], && \text{(19i)}\\
&t_{k,m}[n] > 0, \quad \forall k\in\mathcal{K},\ m\in\mathcal{M},\ n\in\mathcal{N}, && \text{(19j)}
\end{aligned}$$

where $\lambda_{k,n} = s_{k,0}[n]\kappa X_{k,n}D_{k,n}f_{k,n}^2$ and $\mu_{k,m,n} = s_{k,m}[n]p_k D_{k,n}$. Note that the objective function of problem **P2.2.1** is the upper bound of the objective function of problem **P2.2**. Since the problem **P2.2.1** is a convex problem and the UAVs' trajectories $\mathbf{Q}$ are Markov processes, we utilize the convex optimization tool CVX to solve it. Moreover, similar to [13], we utilize the successive convex approach (SCA) to iteratively find the optimal solution. Specifically, in each iteration, the local point $\{\mathbf{q}_m^{\text{local}}[n]\}$ is the solution of the previous iteration. When the change value of the optimal solutions in two adjacent iterations is less than the error threshold, the iteration ends.

C. ITERATIVE ALGORITHM AND ANALYSIS

Then, we propose an iterative optimization algorithm to alternately optimize $\mathbf{S}$ and $\mathbf{Q}$. The iterative algorithm is shown in Algorithm 2.

Moreover, step 3 can obtain the near-optimal $\{s_{k,m}[n]\}$ under fixed $\{\mathbf{q}_m[n]\}$ and step 4 can obtain the optimal $\{\mathbf{q}_m[n]\}$ under a fixed $\{s_{k,m}[n]\}$. Thus, a series of non-increasing objective function values can be obtained. Therefore, according to the block coordinate descent method [37], the proposed algorithm can converge to a sub-optimal solution. In the JOST-Algorithm, the complexity comes from solving the **P2.1.1**, reconstructing offloading strategy variables by the GOSVR-Algorithm and solving the **P2.2.1**. Specifically, the **P2.1.1** and **P2.2.1** are solved by the CVX, which utilizes the primal-dual infeasible interior point method. According to the works in [19] and [36], the computational complexity of solving the **P2.1.1** and **P2.2.1** are $O\left((K(M+1)N+1)^3\right)$ and $O\left((2MN+KMN+2)^3\right)$, respectively. $K(M+1)N+1$ and $2MN+KMN+2$ are the numbers of variables in the **P2.1.1** and **P2.2.1**, respectively. Moreover, the computational complexity of the GOSVR-Algorithm is $O(MN+KMN)$. Hence, the computational complexity of the JOST-Algorithm can be calculated as $O\left(L_1\left(n_1^3+MN+KMN+L_2\,n_2^3\right)\right)$, where n_1 and n_2 are the number of variables in the **P2.1.1** and **P2.2.1**, respectively. The L_1 and L_2 are the total number of iterations of the JOST-Algorithm and the number of iterations of the problem **P2.2.1**, respectively.

Algorithm 2 JOST-Algorithm: Alternately Optimize Offloading Strategies and UAVs' Trajectories

```
1: Initialize $\{s_{k,m}[n], \mathbf{q}_m[n]\}^0$, calculate the weighted sum
   $we^0$ and set the iteration number $l = 0$, $\mathbf{q}_m^{\text{local}}[n] = \mathbf{q}_m^0[n]$
   and the tolerance error $\varepsilon = 10^{-4}$.
2: repeat
3:     Solve the problem P2.1.1 with given $\{\mathbf{q}_m[n]\}^l$
       and obtain the integers $\{s_{k,m}^*[n]\}$ by the GOSVR-Algorithm
4:     Obtain the optimal solutions $\{\mathbf{q}_m^*[n]\}$ by solving the
       problem P2.2.1 with given $\{s_{k,m}^*[n]\}$ and calculate the
       weighted sum $we^*$.
5:     $l = l + 1$
6:     $\{s_{k,m}[n], \mathbf{q}_m[n], we\}^l = \{s_{k,m}^*[n], \mathbf{q}_m^*[n], we^*\}$
7: until $\left|we^l - we^{l-1}\right| \leq \varepsilon$
```

Moreover, the process of obtaining the initial solution of the JOST-Algorithm is as follows. We first draw a circle with a determinate radius and its center is the origin of the coordinate system, and select the initial point of each UAV's trajectory on the circle. Then, we define the above circle starting from the initial point of each UAV as each UAV's trajectory, and divide the circular trajectory from the initial point into equal N segments. The position of each UAV at the n-th time slot corresponds to the initial point of the n-th segment in the circular trajectory. In addition, the flight directions of the UAVs are the same. Thus, collisions between UAVs can be avoided. Moreover, we randomly generate initial computing offloading strategies. Subsequently, we verify whether the initial trajectories and computing offloading strategies met the constraints. If the constraints are met, the process is completed. If the constraints are not met, we change the radius of the circle or computing offloading strategies, and then verify them.

IV. SIMULATION RESULTS

In this section, simulation results are presented to evaluate the performance of the JOST-Algorithm. Denote that the MATLAB is used for the numerical simulations. There are $K = 20$ UEs, which are randomly distributed in the system. Moreover, based on the typical parameter settings in [13], [20] and [38], we set other related system parameters as follows: $N = 50$, $T = 10$ s, $B_{total} = 20$ MHz, $H = 100$ m, $V_{max} = 50$ m/s, $V_{min} = 3$ m/s, $\kappa = 10^{-28}$, $\alpha_0 = -50$ dB, $F = 1.2$ Gcps, $d_{min} = 10$ m, $D_k = 2$ Mbits and $p_k = 10$ dBm, $\forall k \in \mathcal{K}$, $X_{k,n} = 10^3, \forall k \in \mathcal{K}, n \in \mathcal{N}$, and $N_0 = -174$ dBm/Hz. In order to make the weighted ECs of the UEs and the weighted ECs of the UAVs on the order of magnitude as close as possible, and thus to optimize the ECs of both UEs and UAVs better, we compare the UE's local computing energy consumptions with the UAVs' flight energy consumptions under the initial trajectories and set $w_k = 10^3$, $w_m = 10^{-3}$. For convenience, we mainly investigate the typical multi-UAV scenario where $M = 2$, namely, there are two UAVs. Specifically, the initial and final points of UAV 1 and UAV 2 are $\mathbf{q}_1 = [-50, 0, 100]$ and $\mathbf{q}_2 = [50, 100, 100]$. In the section, we also compare the algorithm performances under different numbers of UAVs.

A. PERFORMANCE OF THE JOST-ALGORITHM

To evaluate the performance of the JOST-Algorithm, we compare it with other algorithms. Specifically, the reference algorithms as performance baselines mainly include the comparing algorithm (CA) where based on the optimization ideas in the work [20], the linear computing offloading strategy variables and the UAVs' trajectories are iteratively optimized, and finally the computing offloading strategy variables are restructured into integers by the GOSVR-Algorithm; the joint offloading strategy and trajectory optimizations without UAV optimization (JOSTWUO) where only the maximal EC among all UEs' is minimized; the trajectory optimization with positive offloading strategies (TOWPOS) where only the UAVs' trajectories are optimized to minimize the weighted sum under a random and positive offloading strategies (the positive offloading strategies mean that UEs tend to offload computing tasks to UAVs rather than local computing); the offloading strategy optimization (OSO) where only the computing offloading strategies are optimized to minimize the weighted sum under the initial circular trajectory; the none optimization with positive offloading strategies (NOWPOS) where there is not optimization but with a random and positive offloading strategies. Moreover, we utilize the relaxed lower bound (RLB) to reflect the effectiveness of the proposed algorithm. The RLB is obtained by solving a convex RLB problem. Specifically, in the RLB problem, the computing offloading strategy variables are continuous variables between 0 and 1, the transmission rate of the UE k is changed from $B \log_2(1 + p_k g_0/(N_0\, B \|\mathbf{q}_m[n] - \mathbf{u}_k\|^2))$ to its RLB $B \log_2\left(1 + p_k g_0/(N_0\, BH^2)\right)$, and the maximal energy consumption among all UAVs is set to the minimal flight energy consumption under the feasible range of the UAV's speed.

In Fig. 2, we plot the convergence performance of the JOST. It can be seen that the JOST can converge in two iterations under different T, thus showing good convergence performance. Moreover, as the total task time T becomes larger, the weighted sum of EC becomes smaller, which indicates that there is a tradeoff relationship between task delay and EC. In addition, equally interval increases in total task time T does not correspond to equally interval decreases in the weighted sum of ECs, which shows that the relationship between the total task time and the weighted sum of ECs is not a simple linear relationship.

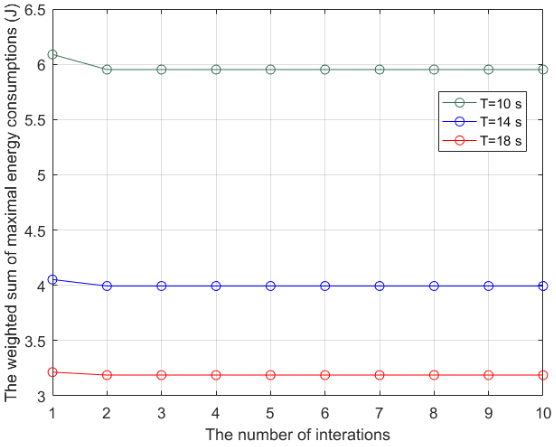


FIGURE 2. **Convergence performance of the JOST-Algorithm under different *T*, with $U_{max} = 5$.**

The weighted sum of maximal ECs under different algorithms and the weighted maximal EC versus the number of UEs are presented in Fig. 3 and Fig. 4, respectively. It is obvious that the JOST is significantly better than the NOWPOS, presenting the tremendous benefits the UEs and

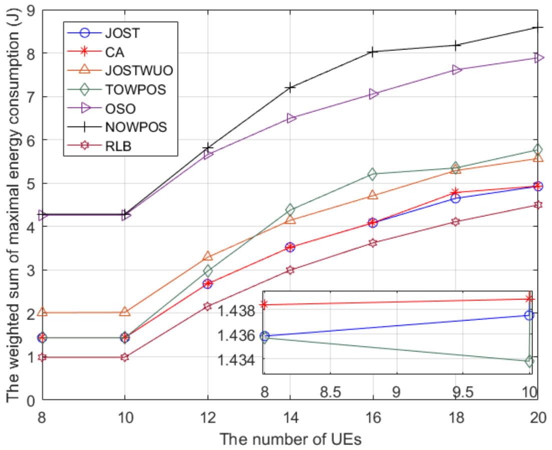


FIGURE 3. **The weighted sum of maximal EC versus the number of UEs under different algorithms, with $U_{max} = 5$.**

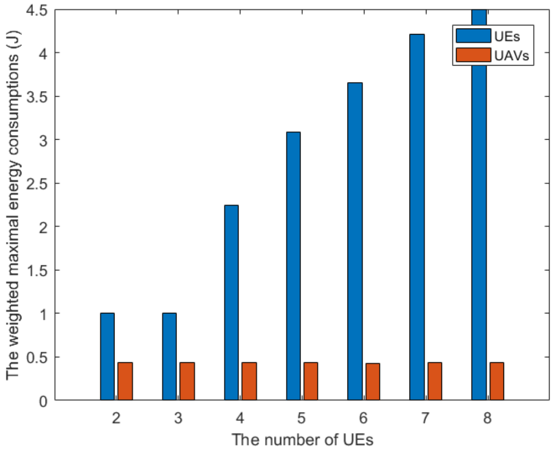


FIGURE 4. The weighted maximal EC versus the number of UEs under the JOST-Algorithm, with $U_{max} = 5$.

UAVs obtained by jointly optimizing offloading strategies and trajectories. It is clear that the weighted sum of the JOST always outperforms that of JOSTWUO, which is because only optimizing the UEs' ECs makes the UAVs tend to be closer to all UEs in the system, which make UAVs consume more energy. In addition, the weighted sum of ECs of the OSO is always higher than that of the TOWPOS, indicating that the efficiency of trajectory optimization is higher than that of computing offloading strategy optimization. This is because the computing offloading strategies mainly affect the ECs of the UEs, and the trajectories of the UAVs affect not only the ECs of the UEs but also the ECs of the UAVs. It can be seen that the performance of the JOST is relatively similar to that of the CA. It shows that the proposed JOST can obtain the performance which is similar to the algorithm based on the existing related work [20], which reflects the effectiveness of the proposed algorithm. In addition, it can be seen that the performance of the JOST can outperform that of the CA in some cases, which also shows the effectiveness of the JOST. Moreover, because the CA is an iterative optimization algorithm, and the optimization problems in the iterative process are **P2.1.1** and **P2.2.1**. Then the computational complexity of the CA is $O\left(L_3\left(n_1^3+L_4\, n_2^3\right)+MN+KMN\right)$, where L_3 and L_4 are the total number of iterations of the CA and the number of iterations of the trajectory optimization, respectively. It can be seen that the computational complexity of the CA and that of the JOST is similar. It is clear that the gap between the weighted sum obtained by the JOST and that obtained by the RLB is not large. In addition, as the number of UEs increases, the change in the gap between the JOST and the RLB is small. The above results verify the validity of the proposed JOST.

Moreover, it can be seen that when the number of UEs is less than MU_{max}, the gap between the JOST and the TOWPOS is small. This shows that when the computing resources in the system are sufficient, the weighted sum are mainly affected by the UAVs' trajectories. Since the TOWPOS and the JOST have different initial offloading strategies, there is no absolute relationship between the weighted sum obtained by the TOWPOS and the JOST, namely, the weighted sum obtained by the JOST may be smaller than that obtained by the TOWPOS, and vice versa. Under the condition of sufficient computing resources, since the performances of the JOST and the TOWPOS are close and the TOWPOS only optimizes the UAVs' trajectories, the TOWPOS is a better choice.

While the the number of UEs is larger than MU_{max}, the performance of the JOST is significantly better than that of other algorithms without the CA, which indicates that when the computing resources are insufficient, the joint optimization of offloading strategies and trajectories can obtain great performance improvement. As can be seen from Fig. 4, when the number of UEs changes from 10 to 12, the weighted maximal EC among UEs suddenly becomes large, while the weighted maximal EC among UAVs changes slightly. This indicates that UEs' ECs are sensitive to the quantitative relationship between the UEs and computing resources.

Fig. 5 shows that the weighted sum of maximal ECs among UEs and UAVs versus the number of UAVs. In particular, we define the coordinates of the initial and final points of the UAV 3 and the UAV 4 as $\mathbf{q}_3 = \{50, 0, 100\}$ and $\mathbf{q}_4 = \{-50, 100, 100\}$, respectively. In general, the performance of the JOST always outperforms that of other baseline algorithms without the CA and the TOWPOS. It can be seen that there is no absolute performance relationship between the JOST and the CA, namely, the JOST does not always outperform the CA, and vice versa. Intuitively, the performance relationship between the JOST and the CA is affected by other system parameters (e.g., the numbers of UEs and UAVs). However, under a variety of different system parameters, the performance of the JOST is generally better than that of the CA, which reflects the effectiveness of the proposed algorithm. It can be seen that when the computing resources are insufficient (e.g., $M \leq 3$ in Fig. 5(a)), the performance gap between the JOST and the RLB is small. As the number

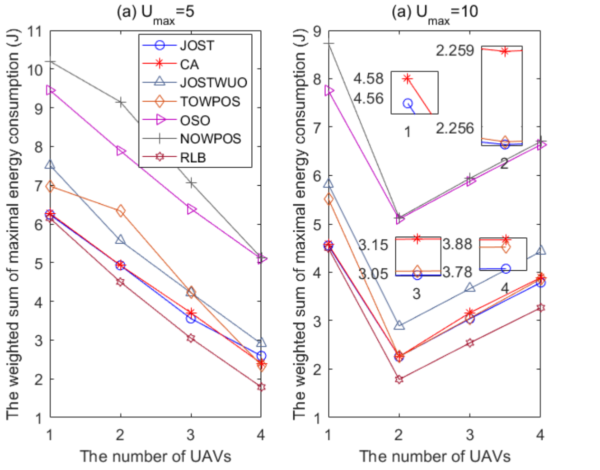


FIGURE 5. The weighted sum of maximal ECs among UEs and UAVs versus the number of UAVs, with $F = 1.2$ Gbps.

of UAVs increases (i.e., the number of computing resources increases), the performance gap between the JOST and the RLB is gradually enlarged. This shows that the JOST can show good performance when computing resources are insufficient. Moreover, it can be seen from Fig. 5, when the computing resources are sufficient, the performance relationship between the JOST and the TOWPOS satisfies the conclusion obtained in Fig. 3, thus further verifying the correctness of the conclusion. It is clear that the performance of multiple UAVs is better than that of single UAV. In addition, under $U_{max} = 10$, when the number of UAVs is increased from 2, the weighted sum appears to increase. This shows that when the communication resources are limited, more UAVs may lead to reduced performance. This is because the increase in the number of UAVs leads to a decrease in the bandwidths of the sub-channels allocated to the UEs, thus reducing the transmission rate and increasing UEs' ECs.

Moreover, in Fig. 5(a), we can observe that the performance relationship between the JOSTWUO and the TOWPOS is not unique. This is because when the number of computing resources is less than the number of UEs (i.e., $M \leq 3$), the performances of the JOSTWUO and the TOWPOS are affected by not only the number of UAVs but also the number of UEs. It can be seen from Fig. 3 that when $M = 2$ and $K = 12$, the performance of the TOWPOS is better than that of the JOSTWUO, thus verifying the above view. However, when the number of computing resources is equal to or greater than the number of UEs (e.g., $M \geq 2$ in Fig. 5(b)), the performance of the TOWPOS is always better than that of the JOSTWUO. This is because the main factors that affect the performance at this time is the UAVs' trajectories, which makes the performance of the TOWPOS close to that of the JOST.

B. FAIRNESS ANALYSIS OF THE JOST-ALGORITHM

Fig. 6 depicts the trajectories obtained by the JOST and the JOSTWUO. The black points in the figure represent UEs. It can be seen from the figure that compared to the UEs' trajectories obtained by the JOSTWUO, the trajectories of UAV 1 and UAV 2 obtained by the JOST are more symmetrical, which makes fairer EC between UAVs. In addition, it is clear that UAVs under the JOSTWUO tend to be closer to each UE, which makes the UAVs consume more energy. However, the UAVs under the JOST have made compromises between approaching UEs and reducing their own ECs. Although it may increase UEs' EC, it can reduce the ECs of UAVs and extend the UAVs' cruising distances.

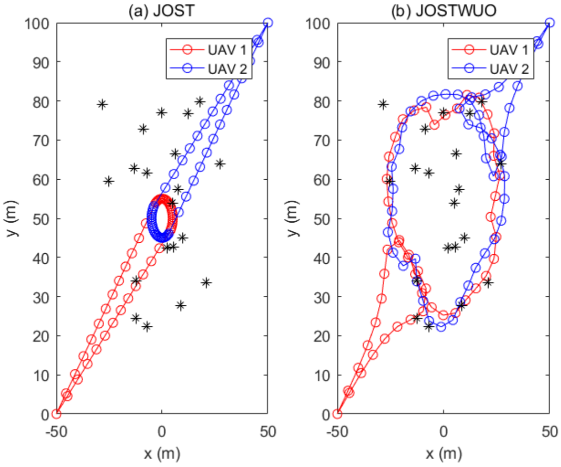


FIGURE 6. Trajectories obtained by the JOST-Algorithm and JOSTWUO-Algorithm, with $U_{max} = 5$.

In Fig. 7 and Fig. 8, we compare the maximal, the minimal and the average ECs of all UEs and UAVs, respectively. In Fig. 7, compared to the TOWPOS, the OSO and the NOWPOS, the maximal EC of the JOST is significantly smaller, and the gaps between the maximal EC, the minimum EC and the average EC are also smaller. This indicates that JOST can effectively reduce the ECs of UEs while guaranteeing the EC fairness of UEs. In particular, the maximal EC obtained by the JOSTWUO is smaller than that obtained by the JOST. This is because the JOSTWUO only focuses on the minimization of the maximal EC among UEs, and the JOST consider not only the UEs but also the UAVs, which further shows that

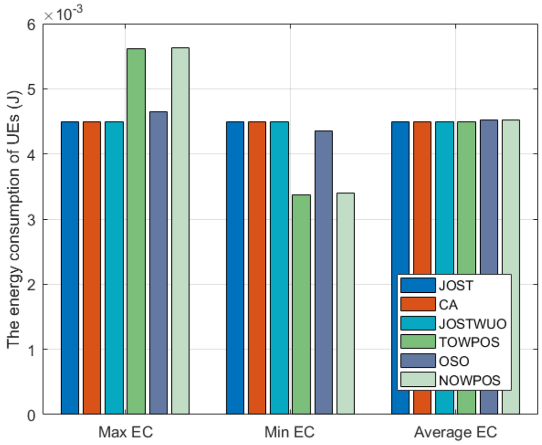


FIGURE 7. Fairness Comparison for UEs, with $U_{max} = 5$.

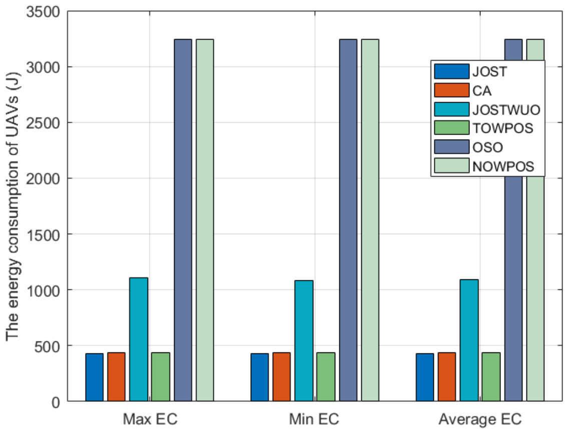


FIGURE 8. Fairness Comparison for UAVs, with $U_{max} = 5$.

optimizing the ECs of the UAVs requires energy sacrifice of UEs.

In Fig. 8, compared with the JOSTWUO, the TOWPOS, the OSO, and the NOWPOS, the maximal EC of the JOST is smaller, and the gaps between the maximal EC, minimum EC, and the average EC are also smaller. This shows that the JOST can effectively guarantee the fairness of UAVs' ECs. In particular, the maximal EC of the JOSTWUO is significantly larger than that of the JOST. Combining the result with Fig. 7, we can know that compared with the JOSTWUO, the JOST can effectively reduce the maximal EC among UAVs with the small cost of the maximal EC among UEs. Moreover, combined with Fig. 7 and Fig. 8, the maximal ECs, minimum ECs and average ECs of UEs and UAVs obtained by the JOST are similar to those obtained by the CA, showing the effectiveness of the proposed algorithm in ensuring the EC fairness of both UEs and UAVs.

V. CONCLUSION

In this paper, we have investigated the multi-UAV enabled MEC system and minimized the weighted sum of the maximal EC among the UEs and the maximal EC among the UAVs. The original problem was divided into two sub-problems and we solved them one by one. Then, we proposed an iterative optimization algorithm to alternately optimize offloading strategies and UAVs' trajectories, and finally obtained a sub-optimal solution of the upper bound problem. Simulation results show that the proposed algorithm can effectively guarantee the EC fairness of UEs and UAVs. Moreover, the performance of multiple UAVs is better than that of single UAV. In addition, when the communication resources are limited, the increase in the number of UAVs may lead to a decrease in performance. In our future work, we will study the trade-off problem between EC minimization and fairness for both UEs and UAVs in the system. The solution of the problem can make the whole system operate with lower and fairer ECs. Moreover, we will study the trade-off between the energy consumption and the number of computed bits at servers by investigating the computation efficiency maximization in edge computing systems.

REFERENCES

[1] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb, "Survey on multi-access edge computing for Internet of Things realization," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2961–2991, Jun. 2018.

[2] S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani, and R. Buyya, "Cloud-based augmentation for mobile devices: Motivation, taxonomies, and open challenges," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 337–368, 1st Quart., 2014.

[3] X. Fan, C. Xiang, C. Chen, P. Yang, L. Gong, X. Song, P. Nanda, and X. He, "BuildSenSys: Reusing building sensing data for traffic prediction with cross-domain learning," *IEEE Trans. Mobile Comput.*, early access, Feb. 28, 2020, doi: 10.1109/TMC.2020.2976936.

[4] C. Xiang, P. Yang, C. Tian, L. Zhang, H. Lin, F. Xiao, M. Zhang, and Y. Liu, "CARM: Crowd-sensing accurate outdoor RSS maps with error-prone smartphone measurements," *IEEE Trans. Mobile Comput.*, vol. 15, no. 11, pp. 2669–2681, Nov. 2016.

[5] J. Zheng, Y. Cai, Y. Wu, and X. Shen, "Dynamic computation offloading for mobile cloud computing: A stochastic game-theoretic approach," *IEEE Trans. Mobile Comput.*, vol. 18, no. 4, pp. 771–786, Apr. 2019.

[6] Y. Wu, K. Ni, C. Zhang, L. P. Qian, and D. H. K. Tsang, "NOMA-assisted multi-access mobile edge computing: A joint optimization of computation offloading and time allocation," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12244–12258, Dec. 2018.

[7] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.

[8] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, 3rd Quart., 2017.

[9] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks," *IEEE Signal Process. Mag.*, vol. 31, no. 6, pp. 45–55, Nov. 2014.

[10] X. Diao, J. Zheng, Y. Cai, Y. Wu, and A. Anpalagan, "Fair data allocation and trajectory optimization for UAV-assisted mobile edge computing," *IEEE Commun. Lett.*, vol. 23, no. 12, pp. 2357–2361, Dec. 2019.

[11] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1123–1152, 2nd Quart., 2016.

[12] J. Wang, C. Jiang, Z. Wei, C. Pan, H. Zhang, and Y. Ren, "Joint UAV hovering altitude and power control for space-air-ground IoT networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1741–1753, Apr. 2019.

[13] 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.

[14] 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.

[15] C. Zhan, Y. Zeng, and R. Zhang, "Energy-efficient data collection in UAV enabled wireless sensor network," *IEEE Wireless Commun. Lett.*, vol. 7, no. 3, pp. 328–331, Jun. 2018.

[16] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2329–2345, Apr. 2019.

[17] B. Li, Z. Fei, and Y. Zhang, "UAV communications for 5G and beyond: Recent advances and future trends," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2241–2263, Apr. 2019.

[18] J. Wang, C. Jiang, Z. Han, Y. Ren, R. G. Maunder, and L. Hanzo, "Taking drones to the next level: Cooperative distributed Unmanned-Aerial-Vehicular networks for small and mini drones," *IEEE Veh. Technol. Mag.*, vol. 12, no. 3, pp. 73–82, Sep. 2017.

[19] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, "Computation rate maximization in UAV-enabled wireless-powered mobile-edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1927–1941, Sep. 2018.

[20] Q. Hu, Y. Cai, G. Yu, Z. Qin, M. Zhao, and G. Y. Li, "Joint offloading and trajectory design for UAV-enabled mobile edge computing systems," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1879–1892, Apr. 2019.

[21] S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2049–2063, Mar. 2018.

[22] J. Hu, M. Jiang, Q. Zhang, Q. Li, and J. Qin, "Joint optimization of UAV position, time slot allocation, and computation task partition in multiuser aerial mobile-edge computing systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 7231–7235, Jul. 2019.

[23] M. Hua, Y. Wang, C. Li, Y. Huang, and L. Yang, "UAV-aided mobile edge computing systems with one by one access scheme," *IEEE Trans. Green Commun. Netw.*, vol. 3, no. 3, pp. 664–678, Sep. 2019.

[24] J. Zhang, L. Zhou, Q. Tang, E. C.-H. Ngai, X. Hu, H. Zhao, and J. Wei, "Stochastic computation offloading and trajectory scheduling for UAV-assisted mobile edge computing," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3688–3699, Apr. 2019.

[25] X. Hu, K.-K. Wong, K. Yang, and Z. Zheng, "UAV-assisted relaying and edge computing: Scheduling and trajectory optimization," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4738–4752, Oct. 2019.

[26] Y. Du, K. Yang, K. Wang, G. Zhang, Y. Zhao, and D. Chen, "Joint resources and workflow scheduling in UAV-enabled wirelessly-powered MEC for IoT systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 10187–10200, Oct. 2019.

[27] Y. Liu, K. Xiong, Q. Ni, P. Fan, and K. B. Letaief, "UAV-assisted wireless powered cooperative mobile edge computing: Joint offloading, CPU control, and trajectory optimization," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2777–2790, Apr. 2020.

[28] Z. Yang, C. Pan, K. Wang, and M. Shikh-Bahaei, "Energy efficient resource allocation in UAV-enabled mobile edge computing networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4576–4589, Sep. 2019.

[29] Y. Wang, Z.-Y. Ru, K. Wang, and P.-Q. Huang, "Joint deployment and task scheduling optimization for large-scale mobile users in multi-UAV-enabled mobile edge computing," *IEEE Trans. Cybern.*, early access, Sep. 11, 2019, doi: 10.1109/TCYB.2019.2935466.

[30] J. Wang, K. Liu, and J. Pan, "Online UAV-mounted edge server dispatching for Mobile-to-Mobile edge computing," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1375–1386, Feb. 2020.

[31] L. Yang, H. Yao, J. Wang, C. Jiang, A. Benslimane, and Y. Liu, "Multi-UAV enabled load-balance mobile edge computing for IoT networks," *IEEE Internet Things J.*, early access, Feb. 4, 2020, doi: 10.1109/JIOT.2020.2971645.

[32] Y. Xu and S. Mao, "A survey of mobile cloud computing for rich media applications," *IEEE Wireless Commun.*, vol. 20, no. 3, pp. 46–53, Jun. 2013.

[33] J. Du, L. Zhao, J. Feng, and X. Chu, "Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee," *IEEE Trans. Commun.*, vol. 66, no. 4, pp. 1594–1608, Apr. 2018.

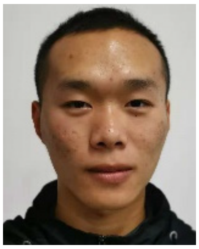
[34] G. T. Ross and R. M. Soland, "A branch and bound algorithm for the generalized assignment problem," *Math. Program.*, vol. 8, no. 1, pp. 91–103, Dec. 1975.

[35] M. Grant and S. Boyd. (Sep. 2013). *CVX: MATLAB Software for Disciplined Convex Programming, Version 2.0 Beta*. [Online]. Available: http://cvxr.com/cvx

[36] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

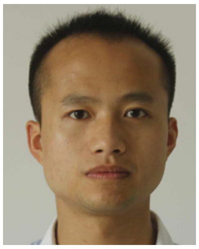
[37] D. P. Bertsekas, "Nonlinear programming," *J. Oper. Res. Soc.*, vol. 48, no. 3, p. 334, 1997.

[38] J. Zhang, X. Hu, Z. Ning, E. C.-H. Ngai, L. Zhou, J. Wei, J. Cheng, and B. Hu, "Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2633–2645, Aug. 2018.



XIANBANG DIAO received the B.S. degree from the College of Communications Engineering, Chongqing University, Chongqing, China, in 2017, and the M.S. degree from the College of Communications Engineering, Army Engineering University of PLA, Nanjing, China, in 2019, where he is currently pursuing the Ph.D. degree. His current research interests include multi-access edge computing, unmanned aerial vehicle communications, and optimization techniques.

MENG WANG received the M.S. degree from the College of Educational Science, Nanjing Normal University, Nanjing, China, in 2009. He is currently a Lecturer at the College of Communications Engineering, Army Engineering University of PLA. His current research interests include computer science and multi-access edge computing.



JIANCHAO ZHENG received the B.S. degree in communications engineering and the Ph.D. degree in communications and information systems from the College of Communications Engineering, PLA University of Science and Technology, Nanjing, China, in 2010 and 2016, respectively. From 2015 to 2016, he was a Visiting Scholar with the Broadband Communications Research Group, Department of Electrical and Computer Engineering, University of Waterloo, Canada. He was also an Assistant Professor with the College of Communications Engineering, Army Engineering University of PLA. He is currently an Assistant Professor at the National Innovation Institute of Defense Technology, Academy of Military Sciences of PLA. His research interests include interference mitigation techniques, green communications and computing networks, game theory, learning theory, and optimization techniques. He has authored or coauthored several papers in international conferences and reputed journals in these research areas.

YUEMING CAI received the B.S. degree in physics from Xiamen University, Xiamen, China, in 1982, and the M.S. degree in micro-electronics engineering and the Ph.D. degree in communications and information systems from Southeast University, Nanjing, China, in 1988 and 1996, respectively. He is currently a Full Professor at the College of Communications Engineering, Army Engineering University of PLA. His current research interests include cooperative communications, signal processing in communications, wireless sensor networks, and physical layer security.