UAV-Enabled Mobile Edge Computing with Binary Computation Offloading and Energy Constraints

Changyuan Xu, Cheng Zhan*, Jingrui Liao, Bin Zeng

School of Computer and Information Science, Southwest University, China
xcy202009@email.swu.edu.cn, zhanc@swu.edu.cn, liaojingrui@email.swu.edu.cn, alison@email.swu.edu.cn

Abstract

Mobile edge computing (MEC) has been considered to provide computation services near the edge of mobile networks, while the unmanned aerial vehicle (UAV) is becoming an important integrated component to extend service coverage. In this paper, we consider a UAV-enabled MEC with binary computation offloading and energy constraints, where an energy-limited UAV is employed as an aerial edge server and each task of devices is either executing locally or offloading to the aerial edge server as a whole. To provide fairness among different ground devices, we aim to maximize the minimum computation throughput among all devices via the joint design of computing mode selection and UAV trajectory as well as resource allocation. The optimization problem is formulated as a mixed-integer non-linear problem consisting of binary variables, which is difficult to tackle. By employing deductive penalty function to penalize the effect of non-binary solution, we develop an efficient iterative algorithm to obtain a suboptimal solution via leveraging the penalty successive convex approximation (P-SCA) method and difference of two convex (D.C.) optimization framework, where the algorithm is guaranteed to converge. Extensive simulations are conducted and the results with different system parameters show that the proposed joint design algorithm can improve the computation throughput by about 40% compared to other benchmark schemes.

Keywords: Unmanned aerial vehicle (UAV), mobile-edge computing (MEC), binary computation offloading, penalty successive convex approximation (P-SCA)

1 Introduction

With the rapid development of 5G networks and Internet of Things (IoT), more and more applications and services require low latency and large computation capacity to provide a better experience for users [1]. For example, auto-driving cars, image and video processing, face recognition, real-time online games, etc. However, most of the IoT devices are resources restrained in terms of energy and computation capacity, which leads to difficulty in maintaining endurance or producing instantaneous responses [2]. To tackle such difficulty, mobile edge computing (MEC) has been introduced [3-5] by moving the powerful computation resource to the edge of the network, and the tasks can be executed on the edge side without transmitting data to remote clouds such that the transmission delay can be significantly reduced. With MEC, the IoT devices can offload their computation tasks for timely processing, thereby saving the energy of the devices [6-7]. The work in [8] optimized resource allocation in mobile computing, and proposed a gateway-based edge computing service model to reduce the latency of data transmission. In [9], the application-oriented offloading in heterogeneous networks was studied for mobile cloud computing, where two algorithms were proposed to minimize execution time. In general, MEC consists of two computation task offloading models, i.e., partial and binary computation offloading [10]. Partial offloading partitions each computation task into two parts, one is for locally computing at ground devices and the other is for offloading to edge server. Binary offloading requires that each task is either executing locally or offloading to the edge server as a whole. In practice, it is easy to implement binary offloading that is suitable for IoT tasks that are not partitionable.

On the other hand, new challenges arise due to the limited coverage of static edge servers. Specifically, static edge servers may not support task offloading for remote devices with complex radio environment and long-distance path loss. In addition, it is very expensive and impractical to deploy a large number of static edge servers in a wide area environment. To tackle such challenges, the UAV-enabled MEC system that supports aerial computing is introduced [11-12], where a flying UAV with powerful computation capacity is employed to provide computation offloading opportunities for resource-restrained ground IoT devices. By taking advantage of the UAV's mobility, the UAV can be close enough to IoT devices, line-of-sight (LoS) links are able to be established with a large probability between the UAV and IoT devices such that the communication quality can be enhanced for computation offloading [13].

In [14], the authors addressed the UAV-aided MEC system and minimized the maximum delay for all users via joint offloading and trajectory design. In [15], the authors investigated the security problems over MEC systems with dual UAVs and maximized secure computing capacity for both time division and non-orthogonal multiple access schemes. The cooperation between the access point and UAV in wireless-powered MEC was studied in [16], where the weighted sum of completed task-input bits was maximized. In [17], the authors take the UAV's trajectory into consideration and proposed two path planning algorithms with 3G Communication. The work in [18] proposed a low-complexity iterative algorithm in a UAV-enabled MEC system with

*Corresponding Author: Cheng Zhan; E-mail: zhanc@swu.edu.cn
DOI: 10.53106/160792642022092305003

secure consideration to maximize the secrecy capacity subject to minimum offloading, latency, and total power constraints. The integration of edge computing and relaying with the help of the UAV was investigated in [19], where the weighted sum of energy consumptions for both users and the UAV was minimized. However, the above existing works relied on the partial computation offloading policy, which is difficult to implement in practice.

Unlike the aforementioned studies, we adopt a binary offloading policy in a UAV-enabled MEC system in this paper, which consists of hard combinatorial mode selection and is thus more challenging compared to partial computation offloading. We aim to maximize the computation throughput via the joint design of UAV trajectory and computation offloading as well as resource allocation. Computation throughput is defined as the total number of processed bits computed at both IoT devices and the UAV edge server, which is a measure of the system's computing capability. To the best of our knowledge, the computation throughput maximization problem for UAV-enabled MEC with binary computation offloading is challenging and has not been investigated in prior works. We summarize the main contributions of this paper as follows:

• Firstly, in order to achieve fairness among all ground devices, we maximize the minimum computation throughput among all devices in UAV-enabled MEC with binary computation offloading policy and energy constraints by jointly optimizing UAV trajectory, computation offloading, and computing frequency of devices. The optimization problem is formulated as a mix-integer non-linear optimization problem, which is challenging to solve in general.

• Secondly, we reformulate the problem into a mathematical equivalent by employing a deductive penalty function to penalize the effect of the non-binary solution, and then develop an efficient iterative algorithm to obtain the binary offloading solution via leveraging the penalty successive convex approximation (P-SCA) method and difference of two convex (D.C.) optimization framework, where the algorithm is guaranteed to converge to a local optimal solution satisfying the Karush-Kuhn-Tucker (KKT) conditions.

• Finally, extensive simulations are conducted to verify the performance of the proposed design. Simulation results with different system parameters show the effectiveness of the proposed joint design algorithm, where our proposed method can effectively improve computation throughput by 40% compared to other benchmarks.

The rest of this paper is organized as follows. Section 2 introduces the system model and presents the mathematical formulation of the problem. Section 3 proposes an efficient solution based on P-SCA and D.C. optimization techniques. Section 4 presents the simulation results, and Section 5 concludes the paper.

2 System Model and Problem Formulation

2.1 System Model

As shown in Figure 1, we consider a UAV-enabled MEC system for IoT applications where a flying UAV is employed as an edge server to provide computation service to a set of K ground devices, which conduct certain computation tasks. Denote $\mathcal{K} = \{s_k, 1 \le k \le K\}$ as the set of ground devices, where the horizontal coordinate of device $s_k \in \mathcal{K}$ is represented by $\mathbf{w}_k \in \mathbb{R}^{2\times1}$. Let T be the total time horizon, which corresponds to the completion time requirement for the computation tasks. It is assumed that the UAV flies at a fixed altitude H above ground and its horizontal coordinate varies over time, which is represented by $\mathbf{q}(t) \in \mathbb{R}^{2\times1}, t \in [0,T]$. For the ease of exposition, the time horizon T is discretized equally into N time slots with element slot length δ_t, where δ_t is sufficiently small so that the UAV's location can be assumed to be approximately unchanged within each time slot. Therefore, the UAV trajectory can be approximately represented by the sequence $\{\mathbf{q}[n], 1 \le n \le N\}$, with $\mathbf{q}[n] \triangleq \mathbf{q}(n\delta_t)$ denoting the UAV's horizontal location at time slot n. As such, $\frac{\|\mathbf{q}[n+1]-\mathbf{q}[n]\|}{\delta_t} \le V_{\max}, \forall n$, with $V_{\max}$ denoting the maximum UAV speed. We assume that the UAV would return to its initial location by the end of the time horizon T in order to periodically serve the ground devices, i.e., $\mathbf{q}[1] = \mathbf{q}[N]$.

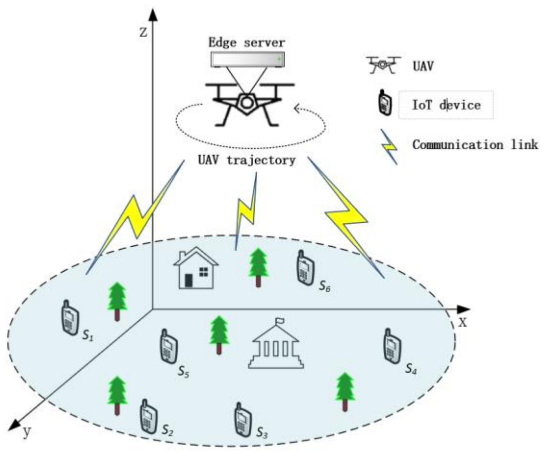


Figure 1. Overview of the UAV-enabled MEC architecture

For the device s_k, we define C_k as the number of CPU cycles needed for computation of 1-bit input data. Denote $f_k[n]$ as the CPU frequency of device s_k at time slot n. Thus, we have $f_k[n] \le f_k^{\max}, \forall n$, where $f_k^{\max}$ is the maximum allowable CPU frequency for s_k. Similarly, we denote $f_{U,k}[n]$ as the allocated CPU frequency for computation tasks offloaded from device s_k at time slot n. Thus we have $\sum_{k=1}^{K} f_{U,k}[n] \le f_U^{\max}$, $\forall n$, where $f_U^{\max}$ is the maximum allowable CPU frequency of UAV. Furthermore, the number of bits computed locally at s_k in time slot n can be calculated by $\frac{f_k[n]\delta_t}{C_k}$. Due to the limited computing capabilities of IoT devices, we assume that the devices adopt a binary task offloading policy [20-21], where a task is either computed locally or offloaded to the UAV for remote computing as a whole. Define $x_k[n] \in \{0,1\}$ as the computation offloading indicator for device s_k, where $x_k[n] = 1$ if the task of s_k is offloaded to the UAV at time slot n and $x_k[n] = 0$ otherwise. To avoid interference among devices during the offloading process, we employ a time division multiple access (TDMA) scheme for computation offloading, where at most, one device offloads the task to the UAV at each time slot. Thus, we have $\sum_{k=1}^{K} x_k[n] \le 1$, ∀n. Similar to [22-23], we assume that the channel between the UAV and each device is mainly a LoS channel. Then the channel power gain between the UAV and device s_k at each time slot n can be modeled as $h_k[n] = \beta_0 d_k^{-\alpha}[n]$, where $\alpha \geq 2$ is the path-loss exponent, β_0 is the average channel power gain at $d_0 = 1$ m, and $d_k(t) = \sqrt{H^2 + \|\mathbf{q}[n] - \mathbf{w}_k\|^2}$ is the distance between the UAV and device s_k at time slot n.

Denote P as the transmit power for device s_k. If offloading, the achievable rate from s_k to the UAV at time slot n can be calculated as $R_k[n] = B\log_2\left(1 + \frac{Ph_k[n]}{\sigma^2}\right) = B\log_2\left(1 + \frac{\gamma_0}{(H^2+\|\mathbf{q}[n]-\mathbf{w}_k\|^2)^{\alpha/2}}\right)$, where B is the channel bandwidth and σ^2 represents for the noise power. As such, $\gamma_0 \triangleq \frac{P\beta_0}{\sigma^2}$ denotes the received signal-to-noise ratio (SNR) at 1 m. For offloaded task computation at the UAV, the information causality-constraints should be imposed, i.e., at any time instant n, the task-input data can only be computed at the UAV if it has already been previously received from the IoT devices. The information causality constraints for offloaded data computation can be expressed as $\delta_t \sum_{n=1}^{N} x_k[n] R_k[n] \geq \delta_t \sum_{n=1}^{N} \frac{f_{U,k}[n]}{C_k}, \forall k$, where $\delta_t \sum_{n=1}^{N} x_k[n] R_k[n]$ represents for the amount of task input data that have been received from devices, while $\delta_t \sum_{n=1}^{N} \frac{f_{U,k}[n]}{C_k}$ denotes those which have been computed at the same time. The energy consumption of IoT device s_k can be constructed by two parts, i.e., the energy consumption related to communication $\sum_{n=1}^{N} x_k[n]\delta_t P$, and the energy consumption related to computation $\sum_{n=1}^{N} \delta_t \kappa_k f_k^3[n]$, where κ_k is the effective capacitance coefficient of device s_k that depends on its processor's chip architecture. To sum up, we have $\sum_{n=1}^{N} x_k[n]\delta_t P + \sum_{n=1}^{N} \delta_t \kappa_k f_k^3[n] \leq E_k^{\max}$, where $E_k^{\max}$ denotes the energy budget of device s_k. Let $\mathbf{v}[n] \triangleq \frac{\|\mathbf{q}[n+1]-\mathbf{q}[n]\|}{\delta_t}$ as the UAV velocity at time slot n, then the energy consumption of rotary-wing UAV at the n-th time slot is calculated as $E^p[n] = \sum_{n=1}^{N} \delta_t (P_0 + \frac{3P_0\|\mathbf{v}[n]\|^2}{U_{tip}^2} + \frac{1}{2} d_0 \rho s A \|\mathbf{v}[n]\|^3) + \sum_{n=1}^{N} \delta_t P_i \left(\sqrt{1 + \frac{\|\mathbf{v}[n]\|^4}{4v_0^4}} - \frac{\|\mathbf{v}[n]\|^2}{2v_0^2}\right)^{1/2}$, where P_0 and P_i are two constants denoting the blade profile and induced power for hovering. v_0 represents the mean rotor induced velocity when hovering, and U_{tip} is the rotor blade's tip speed. Furthermore, s and d_0 represent for the rotor solidity and fuselage drag ratio, respectively. A and ρ represent for the rotor disc area and air density, respectively. Then, we have $E^p[n] \leq E_U^{\max}, \forall$n, where $E_U^{\max}$ denotes the energy budget of the UAV. Note that the binary computation offloading is adopted in this article, then the total computation throughput of device s_k can be given as $\delta_t \sum_{n=1}^{N} \frac{f_{U,k}[n]}{C_k} + \sum_{n=1}^{N} \frac{\delta_t (1-x_k[n]) f_k[n]}{C_k}$. Similar to [8], we assume that time spent on task computation and downloading by the UAV is neglected since the edge server has a much stronger computation capability than the size-constrained IoT devices and the number of bits related to the computation result is very small.

2.2 Problem Statement

In order to achieve fairness among all devices, we aim to maximize the minimum computation throughput among all the ground IoT devices via joint optimizing the resource allocation $\{f_{U,k}[n], f_k(n)\}$ and computation offloading $\{x_k[n]\}$ as well as UAV trajectory $\{\mathbf{q}[n]\}$, subject to the energy budgets of devices and the UAV. Let $\mathbf{Q} \triangleq \{\mathbf{q}[n]\}$, $\mathbf{X} \triangleq \{x_k[n]\}$, $\mathbf{F_k} \triangleq \{f_k[n]\}$, $\mathbf{F_U} \triangleq \{f_{U,k}[n]\}$ and $\eta \triangleq \min_k \{\delta_t \sum_{n=1}^{N} \frac{f_{U,k}[n]}{C_k} + \sum_{n=1}^{N} \frac{\delta_t (1-x_k[n]) f_k[n]}{C_k}\}$. The optimization problem can be formulated as follows:

(P1): $\max_{\mathbf{Q},\mathbf{X},\mathbf{F_k},\mathbf{F_U},\eta} \ \eta$

s. t.

$$\delta_t \sum_{n=1}^{N} \frac{f_{U,k}[n]}{C_k} + \sum_{n=1}^{N} \frac{\delta_t (1-x_k[n]) f_k[n]}{C_k} \geq \eta, \forall k, \quad (1)$$

$$x_k[n] \in \{0,1\}, \forall k, n = 1, \ldots, N, \quad (2)$$

$$\sum_{k=1}^{K} x_k[n] \leq 1, n = 1, \ldots, N, \quad (3)$$

$$\delta_t \sum_{\mathrm{K}=1}^{\mathrm{K}} x_k[n] R_k[n] \geq \delta_t \sum_{\mathrm{n}=1}^{\mathrm{N}} \frac{f_{U,k}[n]}{C_k}, n = 1, \ldots, N, \quad (4)$$

$$f_k[n] \leq f_k^{\max}, \forall k, n = 1, \ldots, N, \quad (5)$$

$$\sum_{k=1}^{K} f_{U,k}[n] \leq f_U^{\max}, n = 1, \ldots, N, \quad (6)$$

$$\sum_{n=1}^{N} x_k[n]\delta_t P + \delta_t \sum_{\mathrm{n}=1}^{\mathrm{N}} \kappa_k f_k^3[n] \leq E_k^{\max}, \forall k, \quad (7)$$

$$E^p[n] \leq E_U^{\max}, \forall n, \quad (8)$$

$$\| \mathbf{q}[n+1] - \mathbf{q}[n] \| \leq V_{max}\delta_t, n = 1, \ldots, N-1, \quad (9)$$

$$\mathbf{q}[1] = \mathbf{q}[N], \quad (10)$$

where (2) and (3) indicate that at most one device is scheduled for communication with UAV in each time slot due to the use of the TDMA.

Note that (P1) is a mixed-integer non-linear optimization problem due to binary constraints in (2) and non-convex constraints in (1), (4) and (8), which are difficult to be solved optimally. In the following, we propose an efficient algorithm to find a suboptimal solution by employing P-SCA and D.C. optimization methods.

3 Proposed Solution

To tackle the binary constraints in (2), we employ the P-SCA method, whose key idea is to add a penalty term that violates the binary constraints to the objective function and then solve the resultant optimization problem by using the SCA technique iteratively. Specifically, we express the constraints in (2) as the intersection of the following regions:

$$0 \leq x_k[n] \leq 1, \forall k, n, \quad (11)$$

$$\sum_{k=1}^{K} \sum_{n=1}^{N} (x_k[n] - x_k^2[n]) \leq 0. \quad (12)$$

Note that constraints in (2) are equivalent to constraints in (11) and (12), since the feasible solution which satisfies the constraints in (2) also satisfies the constraints in (11) and (12) and vice versa. To be specific, it is easy to verify that any feasible point in (2) satisfies constraints in (11) and (12). If constraints in (11) are satisfied, then $0 \leq x_k[n](1-x_k[n]) \leq 1, \forall k, n$ and $\sum_{k=1}^{K}\sum_{n=1}^{N}(x_k[n]-x_k^2[n]) \geq 0$. Combining with (12), we have $\sum_{k=1}^{K}\sum_{n=1}^{N}(x_k[n]-x_k^2[n]) = 0$, and thus $x_k[n](1-x_k[n]) = 0, \forall k, n$. As a result, $x_k[n] \in \{0,1\}, \forall k, n$. Note that the constraints in (11) and (12) are now continuous constraints, and we aim to obtain binary solution for $\mathbf{X}$ in (P1). To this end, we introduce a penalty term to the objective function such that the objective value is penalized if the values of $\mathbf{X}$ are not binary. Thus, the resultant problem is written as

$$\text{(P2):} \quad \max_{\mathbf{Q},\mathbf{X},\mathbf{F_k},\mathbf{F_U},\eta} \eta - \lambda\sum_{k=1}^{K}\sum_{n=1}^{N}(x_k[n]-x_k^2[n])$$
$$\text{s.t.} \quad (1), (3)-(11).$$

Where $\lambda \gg 1$ represents for the penalty factor. It is shown in [24] that when the penalty factor λ is large enough, problem (P2) is equivalent to problem (P1).

By introducing slack variables $\mathbf{Y} \triangleq \{y_k[n]\}$ and $\{\tau[n] \geq 0\}$, where $\tau[n] = \left(\sqrt{1+\frac{\|\mathbf{v}[n]\|^4}{4v_0^4}} - \frac{\|\mathbf{v}[n]\|^2}{2v_0^2}\right)^{1/2}$, we can obtain that $\frac{1}{\tau[n]} = \tau[n]^2 + \frac{\|\mathbf{v}[n]\|^2}{v_0^2}$. Let $E^{\tau}[n] \triangleq \sum_{n=1}^{N}\delta_t(P_0 + \frac{3P_0\|\mathbf{v}[n]\|^2}{U_{tip}^2} + \frac{1}{2}d_0\rho sA\|\mathbf{v}[n]\|^3 + P_i\tau[n])$, (P2) can be written as

$$\text{(P3):} \quad \max_{\mathbf{Q},\mathbf{X},\mathbf{F_k},\mathbf{F_U},\mathbf{Y},\{\tau[n]\},\eta} \eta - \lambda\sum_{k=1}^{K}\sum_{n=1}^{N}(x_k[n]-x_k^2[n])$$
$$\text{s.t.}$$

$$\delta_t\sum_{n=1}^{N}x_k[n]y_k[n] \geq \delta_t\sum_{n=1}^{N}\frac{f_{U,k}[n]}{C_k}, \forall k, \quad \textbf{(13)}$$

$$R_k[n] \geq y_k[n], \forall k, n, \quad \textbf{(14)}$$

$$\tau[n]^2 + \frac{\|\mathbf{v}[n]\|^2}{v_0^2} \geq \frac{1}{\tau[n]}, \forall n, \quad \textbf{(15)}$$

$$E^{\tau}[n] \leq E_U^{\max}, \forall n, \quad \textbf{(16)}$$

$$(1), (3), (5)-(7), (9)-(11).$$

Without loss of optimality to the problem (P3), it is shown that equality holds in constraints in (15). Since otherwise, if there exists one constraint in (15) that is satisfied with strict inequality, then the slack variable $\tau[n]$ can always be decreased to satisfy the equality, and all other constraints are still satisfied without changing the objective value. As such, we can obtain another feasible solution with the same objective value. As a result, problem (P2) is equivalent to the problem (P3). Note that (P3) is still a non-convex optimization problem since the objective function is non-concave and non-convex constraints exist in (1), (13), (14) and (15). However, it is observed that the non-convex terms $x_k[n]y_k[n]$ in (13) and $-x_k[n]f_k[n]$ in (1) can be expressed as the D.C. functions, where D.C. optimization framework [25] can be adopted to tackle such issues. Specifically, $x_k[n]y_k[n] = \frac{1}{2}(x_k[n]+y_k[n])^2 - \frac{1}{2}(x_k^2[n]+y_k^2[n])$ and $-x_k[n]f_k[n] = \frac{1}{2}(x_k^2[n]+f_k^2[n]) - \frac{1}{2}(x_k[n]+f_k[n])^2$. By applying first-order Taylor approximation over the convex terms $(x_k[n]+y_k[n])^2$ and $(x_k^2[n]+f_k^2[n])$ respectively with given local points $x_k^r[n], y_k^r[n], f_k^r[n]$, we have

$$x_k[n]y_k[n] \geq (x_k^r[n]+y_k^r[n])(x_k[n]+y_k[n]) - \frac{1}{2}(x_k^r[n]+y_k^r[n])^2 - \frac{1}{2}(x_k^2[n]+y_k^2[n]) \triangleq \hat{z}_k[n], \quad \textbf{(17)}$$

$$-x_k[n]f_k[n] \geq -\frac{1}{2}(x_k[n]+f_k[n])^2 + \frac{1}{2}(x_k^r[n]^2 + f_k^r[n]^2) + x_k^r[n](x_k[n]-x_k^r[n]) + f_k^r[n](f_k[n]-f_k^r[n]) \triangleq \check{z}_k[n], \quad \textbf{(18)}$$

where $\hat{z}_k[n]$ and $\check{z}_k[n]$ are both concave functions.

Then, for (15), given local points $\tau^r[n]$ and $\mathbf{v}^r[n]$ in the r-th iteration, we have the following inequation by applying the first-order Taylor expansion to approximate the convex terms, i.e., $\tau[n]^2 + \frac{\|\mathbf{v}[n]\|^2}{v_0^2} \geq \tau^r[n]^2 + 2\tau^r[n](\tau[n]-\tau^r[n]) - \frac{\|\mathbf{v}^r[n]\|^2}{v_0^2} + \frac{2}{v_0^2}(\tau^r[n])^T\mathbf{v}[n] \triangleq \varphi^{lb}[n]$, where $\varphi^{lb}[n]$ is now joint concave with respect to $\tau[n]$ and $\mathbf{v}[n]$.

Similarly, the objective function in (P3) is lower bounded by $\eta - \lambda\sum_{k=1}^{K}\sum_{n=1}^{N}(x_k[n]-x_k^2[n]) \geq \eta - \lambda\sum_{k=1}^{K}\sum_{n=1}^{N}\left(x_k[n]-x_k^r[n]^2 - 2x_k^r[n](x_k[n]-x_k^r[n])\right) \triangleq \Upsilon^{lb}$ through applying first-order Taylor approximation over term $x_k^2[n]$. On the other hand, $R_k[n]$ is convex with respect to term $\|\mathbf{q}[n]-\mathbf{w}_k\|^2$, then we can obtain the following lower bound as in [22, 26], i.e.,

$$R_k[n] \geq -A_k^r[n]\left(\|\mathbf{q}[n]-\mathbf{w}_k\|^2 - \|\mathbf{q}^r[n]-\mathbf{w}_k\|^2\right) + B_i^r[n] \triangleq R_k^{lb}[n] \quad \textbf{(19)}$$

Where $A_k^r[n] = \frac{B\gamma_0\left(\frac{\alpha}{2}\right)\log_2 e}{\Omega(\Omega^{\frac{\alpha}{2}}+\gamma_0)}$, $B_i^r[n] = Blog_2(1+\frac{\gamma_0}{\Omega^{\frac{\alpha}{2}}})$, and $\Omega \triangleq H^2 + \|\mathbf{q}^r[n]-\mathbf{w}_k\|^2$. In this case, $R_k^{lb}[n]$ is a concave function with respect to $\mathbf{q}[n]$ while Υ^{lb} is a linear function. By applying the lower bound expressions derived in (17)-(19) as well as Υ^{lb}, (P3) can be approximated as

$$\text{(P4):} \quad \max_{\mathbf{Q},\mathbf{X},\mathbf{F_k},\mathbf{F_U},\mathbf{Y},\{\tau[n]\},\eta} \Upsilon^{lb}$$
$$\text{s.t.}$$

$$\delta_t\sum_{n=1}^{N}\frac{f_{U,k}[n]}{C_k} + \sum_{n=1}^{N}\frac{\delta_t(f_k[n]+\check{z}_k[n])}{C_k} \geq \eta, \forall k, \quad \textbf{(20)}$$

$$\delta_t\sum_{n=1}^{N}\hat{z}_k[n] \geq \delta_t\sum_{n=1}^{N}\frac{f_{U,k}[n]}{C_k}, \forall k, \quad \textbf{(21)}$$

$$R_k^{lb}[n] \geq y_k[n], \forall k, n, \quad \textbf{(22)}$$

$$\varphi^{lb}[n] \geq \frac{1}{\tau[n]}, \forall n, \quad \textbf{(23)}$$

$$(3), (5)-(7), (9)-(11), (16).$$

It can be shown that (P4) is a standard convex optimization problem with a concave objective function and a convex constraint set. With a given penalty parameter λ and feasible points $\{x_k^r[n], y_k^r[n], f_k^r[n], \mathbf{q}^r[n], \tau^r[n]\}$, We are able to solve (P4) efficiently by standard convex optimization techniques or solvers such as CVX [27]. In addition, since the lower bound approximation was applied to transform (P3) into (P4), then the constraint set of (P4) is stricter than that of (P3).

In this case, the optimal solution to (P4) is also feasible to (P3). We give the P-SCA-based algorithm to solve problem (P4), where the details are summarized in Algorithm 1. In particular, (P3) is solved by successively solving (P4) for given feasible points $\{x_k^r[n], y_k^r[n], f_k^r[n], \mathbf{q}^r[n], \tau^r[n]\}$, which is updated at the r-th iteration. The initial UAV trajectory can be set as a simple circular trajectory with maximum speed, and the circle center is the geometric center of all ground IoT devices, such that the devices can be served periodically. Similar to [24], the P-SCA-based algorithm is guaranteed to converge to a stationary point, i.e., satisfying the Karush-Kuhn-Tucker (KKT) conditions of (P3). The overall complexity of Algorithm 1 can be given as $O((KN)^{3.5}\log(1/\epsilon))$ since we solve a standard convex optimization problem in each iteration, where ϵ denotes the solution accuracy [27].

Algorithm 1. P-SCA based algorithm for problem (P3)

1: Initialize $\{\mathbf{q}^0[n],\ \boldsymbol{f}_k^0[n],\ \boldsymbol{x}_k^0[n],\ \boldsymbol{y}_k^0[n],\ \boldsymbol{\tau}^0[n]\}$; Set $\boldsymbol{r} = \mathbf{0}$.

2: **repeat**

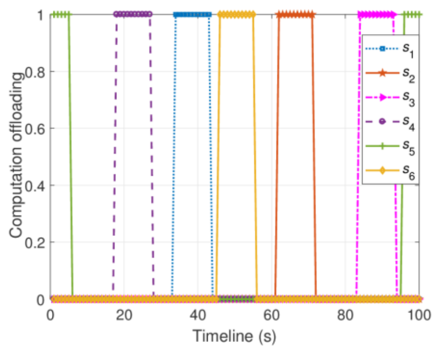
3: With given local points $\{\mathbf{q}^r[n],\ \boldsymbol{f}_k^r[n],\ \boldsymbol{x}_k^r[n],\ \boldsymbol{y}_k^r[n],\ \boldsymbol{\tau}^r[n]\}$, obtain optimal solution $\{\mathbf{q}^*[n],\ \boldsymbol{f}_k^*[n],\ \boldsymbol{x}_k^*[n],\ \boldsymbol{y}_k^*[n],\ \boldsymbol{\tau}^*[n],\ \boldsymbol{f}_{U,k}^*[n]\}$ by solving convex optimization problem (P4) with CVX solver.

4: Update the local points by $\mathbf{q}^{r+1}[n] = \mathbf{q}^*[n], \boldsymbol{x}_k^{r+1}[n] = \boldsymbol{x}_k^*[n], \boldsymbol{y}_k^{r+1}[n] = \boldsymbol{y}_k^*[n], \boldsymbol{f}_k^{r+1}[n] = \boldsymbol{f}_k^*[n], \boldsymbol{\tau}^{r+1}[n] = \boldsymbol{\tau}^*[n], \forall \boldsymbol{k}, \boldsymbol{n}.$

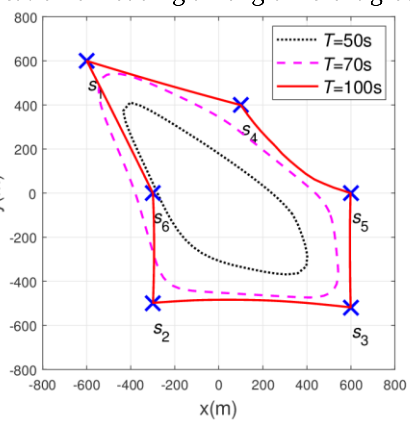
5: Update $\boldsymbol{r} = \boldsymbol{r} + \mathbf{1}$.

6: **until** The objective value of (P3) converges.

4 Simulation Results



(a) Communication offloading among different ground devices



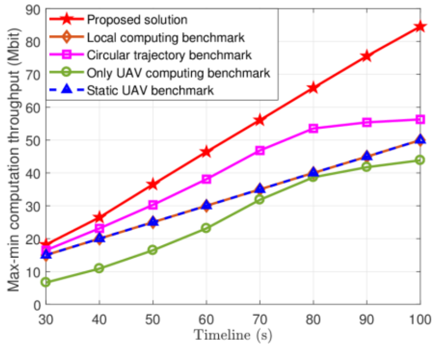
(b) Optimized UAV trajectories with different T

Figure 2. Optimized UAV trajectories and computation offloading

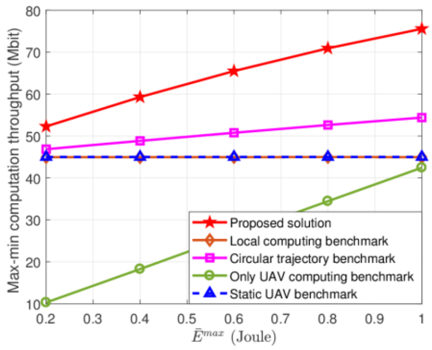
In this section, we present simulations to evaluate the performance of the proposed algorithm. We consider a UAV-enabled MEC system with $K = 6$ IoT devices as shown in Figure 2(b). The ground devices are randomly and uniformly distributed in a $1.6 \times 1.6\ \text{km}^2$ square area. We select computation throughput as the performance metric, which is a measure of the computing capability of a system. We assume that each device has identical energy budget and maximum allowable CPU frequency, i.e., $E_k^{\max} = \bar{E}^{\max}$, $f_k^{\max} = \bar{f}^{\max}$, $f_{U,k}^{\max} = \bar{f}_{U,k}^{\max}$, $\forall k$. Unless otherwise stated, the relevant parameters are set as follows: $B = 1\text{MHz}$, $H = 100$ m, $\sigma^2 = -110\text{dBm}$, $\rho_0 = -50$ dB , $P = 0.1$ W, $\alpha = 2.2$, $V_{\max} = 50$ m/s , $E_U^{\max} = 2 \times 10^5$ Joule , $\bar{f}^{\max} = 0.5$ GHz, $\bar{f}_{U,k}^{\max} = 10$ GHz, $\epsilon = 10^{-4}$, $\delta_t = 1$ s, $\lambda = 10^5, C_k = 10^3, k_k = 10^{-28}, \forall k$.

In Figure 2(a), we show the computation offloading indicator for different ground devices when $T = 100$ s and $\bar{E}^{\max} = 1$ Joule, $E_U^{\max} = 2 \times 10^5$ Joule. It is observed that the communication offloading indicator is either 0 or 1, which means that binary computation offloading policy is adopted, and it can be efficiently obtained by the proposed Algorithm 1. The reason is that the deductive penalty function is introduced in our algorithm to penalize the effect of non-binary solution, and thus each computing task of the devices is either executed locally or completely offloaded to the UAV server. In Figure 2(b), we show the optimized trajectories obtained by Algorithm 1 over different time horizons T when $\bar{E}^{\max} = 1$ Joule and $E_U^{\max} = 2 \times 10^5$ Joule. It can be seen that the UAV exploits its mobility to adjust its trajectory to fly closer to the devices as T increases. The reason is that the quality of air-to-ground channels is better when the UAV is closer to devices, then offloading tasks to the UAV is more energy-efficient than local computing, and thus ground devices prefer to offload computation tasks to the UAV. Figure 3 shows the max-min computation throughput achieved by our proposed algorithm and the following benchmark schemes: 1) Local computing benchmark, where all ground devices only perform local computing; 2) Only UAV computing benchmark, where the devices offload all the tasks to the UAV for computation without local computing, similar as in [16]; 3) Circular trajectory benchmark, where the UAV flies by following a circular centered at the geometric center of all ground devices, similar as [22]; 4) Static UAV benchmark, where the UAV remains static at the geometric center of the devices, similar as [28]. From Figure 3(a), we can see that the computation throughput increases with the increase of T for all the schemes, and the performance gain is more remarkable with larger T since a larger T provides the UAV enough time to move closer to the devices to be served. For example, when $T = 90$ s and $T = 100$ s, the proposed scheme can improve the computation throughput by about 40% compared with other benchmarks. Furthermore, the rate of computation throughput decreases with large T, and the curves are expected to saturate when T is sufficiently large due to the limited device energy. Figure 3(b) shows the max-min computation throughput versus energy budget $\bar{E}^{\max}$ when $T = 90$ s and $E_U^{\max} = 2 \times 10^5$ Joule. We can see that the computation throughput increases as $\bar{E}^{\max}$ increases since more computation offloading and local computing capacities can be provided from the devices. Furthermore, Figure 4 shows the max-min computation throughput versus UAV energy budget $E_U^{\max}$ when $T = 90$ s and $\bar{E}^{\max} = 1$ Joule.

We can also observe that the computation throughput increases as $E_U^{\max}$ increases since the UAV provides more computing resources and edge computing capabilities. By comparing the performance of the proposed solution with that of the circular trajectory benchmark and static UAV benchmark, the advantage of flexible trajectory design is demonstrated. The performance gains among the proposed solution, local computing benchmark and only the UAV computing benchmark demonstrate the flexible computation offloading design.



(a) Max-min computation throughput versus T



(b) Max-min computation throughput versus $\bar{E}^{\max}$

Figure 3. The max-min computation throughput versus time horizon T or energy budget $\bar{E}^{\max}$

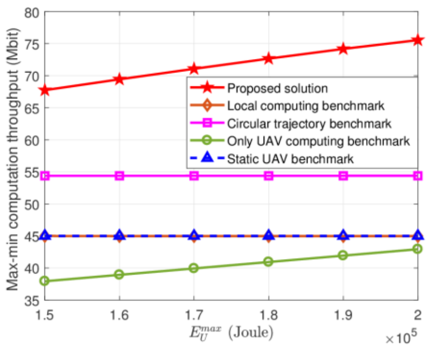


Figure 4. The max-min computation throughput versus UAV energy budget $E_U^{\max}$

5 Conclusion

In this paper, we study the computation throughput maximization problem in a UAV-enabled multi-user MEC system with binary computation offloading scheme. The problem is formulated as a joint optimization of computation offloading and UAV trajectory as well as the computing frequency of devices. We propose an efficient iterative algorithm to obtain a suboptimal KKT solution by employing the P-SCA method and D.C. optimization framework, where the auxiliary penalty function tackles the difficult binary computation mode selection and the D.C. optimization framework tackles the non-convex terms in the optimization problem. Extensive simulation results show that the proposed joint design algorithm can improve the computation throughput by about 40% compared with other benchmark schemes under different network setups.

Acknowledgement

Cheng Zhan is corresponding author for this paper. This work was supported by National Natural Science Foundation of China (No. 62172339). We also thank SouthWest University Big Data Innovation Center for providing the computing resources.

References

[1] W. Saad, M. Bennis, M. Chen, A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems, *IEEE Network*, Vol. 34, No. 3, pp. 134-142, May/June, 2020.

[2] M. Chen, Y. Hao, Task Offloading for Mobile Edge Computing in Software Defined Ultra-Dense Network, *IEEE Journal on Selected Areas in Communications*, Vol. 36, No. 3, pp. 587-597, March, 2018.

[3] Z. Zhou, Q. Wu, X. Chen, Online Orchestration of Cross-Edge Service Function Chaining for Cost-Efficient Edge Computing, *IEEE Journal on Selected Areas in Communications*, Vol. 37, No. 8, pp. 1866-1880, August, 2019.

[4] S. Chen, S. Mei, G. Jia, Y. Li, W. Zhao, KFPA Monocular Ranging Algorithm Design and Application in Mobile edge Computing, *Journal of Internet Technology*, Vol. 22, No. 5 , pp. 1131-1142, September, 2021.

[5] Y.-S. Lin, C.-F. Lai, C.-L. Chuang, X. Ge, H.-C. Chao, Collaborative Framework of Accelerating Reinforcement Learning Training with Supervised Learning Based on Edge Computing, *Journal of Internet Technology*, Vol. 22, No. 2 , pp. 229-238, March, 2021.

[6] C. You, K. Huang, H. Chae, B.-H. Kim, Energy-Efficient Resource Allocation for Mobile-Edge Computation Offloading, *IEEE Transactions on Wireless Communications*, Vol. 16, No. 3, pp. 1397-1411, March, 2017.

[7] X. Chen, L. Jiao, W. Li, X. Fu, Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing, *IEEE/ACM Transactions on Networking*, Vol. 24, No. 5, pp. 2795-2808, October, 2016.

[8] C.-W. Tseng, F.-H. Tseng, Y.-T. Yang, C.-C. Liu, L.-D. Chou, Task Scheduling for Edge Computing with Agile VNFs On-Demand Service Model toward 5G and Beyond, *Wireless Communications and Mobile Computing*, Vol. 2018, pp. 1-13, July, 2018.

[9] F.-H. Tseng, H.-H. Cho, K.-D. Chang, J.-C. Li, T. K. Shih, Application-oriented offloading in heterogeneous networks for mobile cloud computing, *Enterprise*

Information Systems, Vol. 12, No. 4, pp. 398-413, March, 2018.

[10] L. Huang, S. Bi, Y. -J. A. Zhang, Deep Reinforcement Learning for Online Computation Offloading in Wireless Powered Mobile-Edge Computing Networks, *IEEE Transactions on Mobile Computing*, Vol. 19, No. 11, pp. 2581-2593, 1 November, 2020.

[11] F. Zhou, Y. Wu, R. Q. Hu, Y. Qian, Computation Rate Maximization in UAV-Enabled Wireless-Powered Mobile-Edge Computing Systems, *IEEE Journal on Selected Areas in Communications*, Vol. 36, No. 9, pp. 1927-1941, September, 2018.

[12] Z. Yang, C. Pan, K. Wang, M. Shikh-Bahaei, Energy Efficient Resource Allocation in UAV-Enabled Mobile Edge Computing Networks, *IEEE Transactions on Wireless Communications*, Vol. 18, No. 9, pp. 4576-4589, September, 2019.

[13] L. Gupta, R. Jain, G. Vaszkun, Survey of Important Issues in UAV Communication Networks, *IEEE Communications Surveys & Tutorials*, Vol. 18, No. 2, pp. 1123-1152, Secondquarter, 2016.

[14] Q. Hu, Y. Cai, G. Yu, Z. Qin, M. Zhao, G. Y. Li, Joint Offloading and Trajectory Design for UAV-Enabled Mobile Edge Computing Systems, *IEEE Internet of Things Journal*, Vol. 6, No. 2, pp. 1879-1892, April, 2019.

[15] Y. Xu, T. Zhang, D. Yang, Y. Liu, M. Tao, Joint Resource and Trajectory Optimization for Security in UAV-Assisted MEC Systems, *IEEE Transactions on Communications*, Vol. 69, No. 1, pp. 573-588, January, 2021.

[16] X. Hu, K. Wong, K. Yang, Z. Zheng, UAV-Assisted Relaying and Edge Computing: Scheduling and Trajectory Optimization, *IEEE Transactions on Wireless Communications*, Vol. 18, No. 10, pp. 4738-4752, October, 2019.

[17] F.-H. Tseng, T.-T. Liang, C.-H. Lee, L.-D. Chou, H.-C. Chao, A Star Search Algorithm for Civil UAV Path Planning with 3G Communication, *2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, Kitakyushu, Japan, 2014, pp. 942-945.

[18] Y. Zhou, C. Pan, P. L. Yeoh, K. Wang, M. Elkashlan, B. Vucetic, Y. Li, Secure Communications for UAV-Enabled Mobile Edge Computing Systems, *IEEE Transactions on Communications*, Vol. 68, No. 1, pp. 376-388, January, 2020.

[19] X. Hu, K. -K. Wong, Y. Zhang, Wireless-Powered Edge Computing With Cooperative UAV: Task, Time Scheduling and Trajectory Design, *IEEE Transactions on Wireless Communications*, Vol. 19, No. 12, pp. 8083-8098, December, 2020.

[20] Y. Mao, C. You, J. Zhang, K. Huang, K. B. Letaief, A Survey on Mobile Edge Computing: The Communication Perspective, *IEEE Communications Surveys & Tutorials*, Vol. 19, No. 4, pp. 2322-2358, Fourthquarter, 2017.

[21] S. Bi, Y. J. Zhang, Computation Rate Maximization for Wireless Powered Mobile-Edge Computing With Binary Computation Offloading, *IEEE Transactions on Wireless Communications*, Vol. 17, No. 6, pp. 4177-4190, June, 2018.

[22] Q. Wu, R. Zhang, Common Throughput Maximization in UAV-Enabled OFDMA Systems With Delay Consideration, *IEEE Transactions on Communications*, Vol. 66, No. 12, pp. 6614-6627, December, 2018,

[23] C. Zhan, H. Hu, Z. Liu, Z. Wang, S. Mao, Multi-UAV-Enabled Mobile-Edge Computing for Time-Constrained IoT Applications, *IEEE Internet of Things Journal*, Vol. 8, No. 20, pp. 15553-15567, October, 2021.

[24] B. Khamidehi, A. Rahmati, M. Sabbaghian, Joint Sub-Channel Assignment and Power Allocation in Heterogeneous Networks: An Efficient Optimization Method, *IEEE Communications Letters*, Vol. 20, No. 12, pp. 2490- 2493, December, 2016.

[25] B. Soleimani, M. Sabbaghian, Cluster-Based Resource Allocation and User Association in mmWave Femtocell Networks, *IEEE Transactions on Communications*, Vol. 68, No. 3, pp. 1746-1759, March, 2020.

[26] C. Zhan, H. Hu, X. Sui, Z. Liu, D. Niyato, Completion Time and Energy Optimization in the UAV-Enabled Mobile-Edge Computing System, *IEEE Internet of Things Journal*, Vol. 7, No. 8, pp. 7808-7822, August, 2020.

[27] S. Boyd, L. Vandenberghe, *Convex Optimization*, Cambridge, U.K.: Cambridge University Press, 2004.

[28] L. Hu, Y. Tian, J. Yang, T. Taleb, L. Xiang, Y. Hao, Ready Player One: UAV-Clustering-Based Multi-Task Offloading for Vehicular VR/AR Gaming, *IEEE Network*, Vol. 33, No. 3, pp. 42-48, May/June, 2019.

Biographies

Changyuan Xu was born in Sichuan Province, China. He is now a graduate student in the School of Computer and Information Science, Southwest University, Chongqing, China. His research interests include wireless communication, mobile edge computing.

Cheng Zhan received the Ph.D. degree in computer science from the University of Science and Technology of China in 2011. He is currently with the School of Computer and Information Science, Southwest University, China. His research interests include network coding, wireless network optimization, multimedia transmission and distributed storage.

Jingrui Liao was born in Chongqing, China. She is now a graduate student in the School of Computer and Information Science, Southwest University, Chongqing, China. Her research interests include wireless communication, video transmission.

Bin Zeng was born in Jiangxi Province, China. He is now a graduate student in the School of Computer and Information Science, Southwest University, Chongqing, China. His research interests include wireless communication, edge caching.