

RESEARCH

Open Access



Distributed energy-efficient and secure offloading in air-to-ground MEC networks

Wanning Liu, Yitao Xu, Ducheng Wu*, Haichao Wang, Xueqiang Zheng and Xueqiang Chen

*Correspondence:
wuducheng@foxmail.com
College of Communications
Engineering, Army
Engineering University
of PLA, Nanjing, China

Abstract

This paper mainly investigates the energy-efficient and secure offloading problem in air-to-ground Mobile Edge Computing (MEC) networks. The proposed efficient offloading mechanism is as per the requirements of the heterogeneous tasks of ground users. Further, the optimizing offloading rate, offloading object, and channel access jointly formulate system energy consumption and eavesdropping rate minimization. A distributed two-stage offloading scheme is proposed for achieving the sub-optimal solution for the Mixed-integer Nonlinear Programming (MINLP) problem. Finally, simulation results demonstrate that the proposed scheme is superior to several benchmark schemes.

Keywords: MEC, Task offloading, Energy-efficient offloading, Secure offloading

1 Introduction

Mobile Edge Computing (MEC) is a promising technology for future wireless networks as it can provide low-latency and context-aware services for terminal users [1, 2]. With the improvement of software and hardware capabilities, Unmanned Aerial Vehicles (UAVs) have become efficient edge computing node due to easy deployment and good communication quality for ground users [3–5]. However, the UAV network has limited energy [6, 7] and is vulnerable to eavesdropping attacks [8–10]. Thus, designing an energy-efficient and secure offloading mechanism is significant. Hence, we study the energy-efficient and secure offloading in air-to-ground MEC networks by jointly optimizing offloading strategies and resource allocation.

For solving long-distance and backhaul link congestion problem in cloud computing [11], the MEC paradigm deploys cloud servers close to the mobile devices, for instance, by integrating computing servers inside Base Stations (BSs) or Access Points (APs) on the network architecture. Mobile users can utilize both the computing and storage resources of surrounding BSs and idle end devices [12] by cellular communication and Device-to-Device (D2D) communication. Recently, higher mobility and lower cost of UAVs have led to significant research interests in the UAV-assisted MEC systems. UAVs as a computing server can improve communication conditions and service quality to mobile users [13].

By deploying a UAV-assisted MEC system, end devices can efficiently complete delay-sensitive applications. In [14–20], the secure data offloading mechanisms were investigated along with the research on UAV-assisted MEC systems [21–24]. Physical layer security has become one of the constraints when data offloading occurs in the MEC system. Thus, it's critical to design an energy-efficient [25, 26] and secure data offloading framework. However, the main way to avoid eavesdropping is to apply jamming signals or change the transmission power in existing work. And the optimization problem in these works is often difficult to resolve with traditional optimization methods. Hence, to solve the above problem, we exploit secure channel access by channel switching [27–31] and propose a two-stage task and resource assignment method that exploits convex optimization theory [32] and fading memory joint strategy fictitious play with inertia (FM-JSFP) [33] to formulate the problem and resolve it.

The main contributions of this paper are summarized as follows.

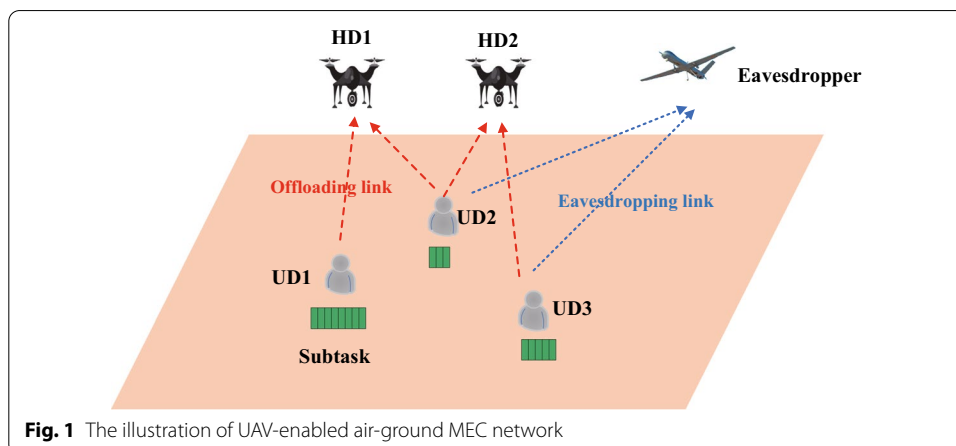
- The designed multi-phase task offloading mechanism based on the time block is as per the MEC framework. Further, optimizing offloading objects, offloading rate, and channel access jointly formulate the system function (i.e., the weighted sum of energy consumption and eavesdropping rate) minimization problem with task delay and computing capacity.
- To solve the non-convex optimization problem, we propose a distributed two-stage source allocation algorithm. The algorithm based on the convex optimization method and FM-JSFP algorithm enables UDs to make sub-optimal decisions. And it does not need a central decision controller.
- Simulation results demonstrate that the proposed scheme improves the system energy-saving and security compared with some benchmark schemes. What's more, it achieves the sub-optimal performance on system function.

2 Related work

Recently, the UAV-enabled MEC systems have investigated in the existing studies. The work presented in [13] considers a resource scheduling mechanism in UAV-enabled MEC systems for Internet of Things Devices (IoTD). Further, UAV provides computing and energy resources to IoTD at the same time. Finally, the energy consumption of the system decreases by multi-resource scheduling. In [21], the authors studied a UAV-assisted MEC system having a moving UAV endowed with computing capabilities to offer sufficient computing resources to mobile users (MUs) and limited local task processing capabilities. The system aims at minimizing the overall energy consumption for the MUs. In [34, 35], the authors considered a wireless communication system having a rotary-wing UAV equipped with a computing server to assist several mobile terminals (MTs). The proposed method effectively reduces the maximum energy consumption among all MTs by transforming the non-convex optimization problem into efficient sub-problems. In [22], the authors studied complement time and energy optimization problem in the UAV-enabled MEC system.

Some of the works focused on multi-UAV-assisted MEC scenarios. In [36], the researchers established a new multi-UAV MEC system. It also presented an efficient two-layer optimization scheme for jointly optimizing the horizontal deployment position of UAVs and task scheduling, intending to minimize the energy consumption of all UAVs. The proposed algorithm has potential applications in large-scale mobile user scenarios. In [5], the authors formulated the sum power minimization problem with latency and coverage constraints as three subproblems and their respective solutions. The study also considered the energy consumption of all User Equipments (UEs) and UAVs as the UAVs are also power constraints. However, the offloading data mechanism for UEs covered by the same UAV is unclear. In [37], the authors focus on the scenario with multiple UAVs having different computing capacities supporting data collection and processing for IoT devices. The Time Division Duplexing (TDD) mode is applied to every IoT devices served by one UAV. Dissimilar to the above traditional optimization methods (i.e., based on the convex optimization theory), [38] proposed a reinforcement learning (RL)-based user association and resource assignment algorithm for addressing the minimum energy consumption problem. However, the transmission mechanism for the user offload data is not provided (i.e., TDD mechanism, FDD mechanism, or others in reality). In [39], system energy consumption minimization problem was studied in the UAV-assisted MEC system. To solve the MINLP problem, a two-stage offloading scheme based on convex optimization and stochastic learning automata (SLA) algorithm was proposed. The simulation results show that the distributed hierarchical mechanism is suitable for solving complex optimization problems.

Studies on security mechanisms for IoT devices in MEC systems have emerged. In [14], the researchers investigated a new MEC framework achieving secure communication for IoT applications. It exploits the key for achieving a reliable connection between the mobile users and the servers. In [15], the authors consider the physical layer security problem of terrestrial MEC systems. Mobile stations optimize system energy consumption under the constraints of anti-eavesdropping. In [16, 17], the authors study secure offloading decisions based on security-awareness in MEC systems. IoT devices prefer servers not attacked by the jammers for obtaining stable and secure services.



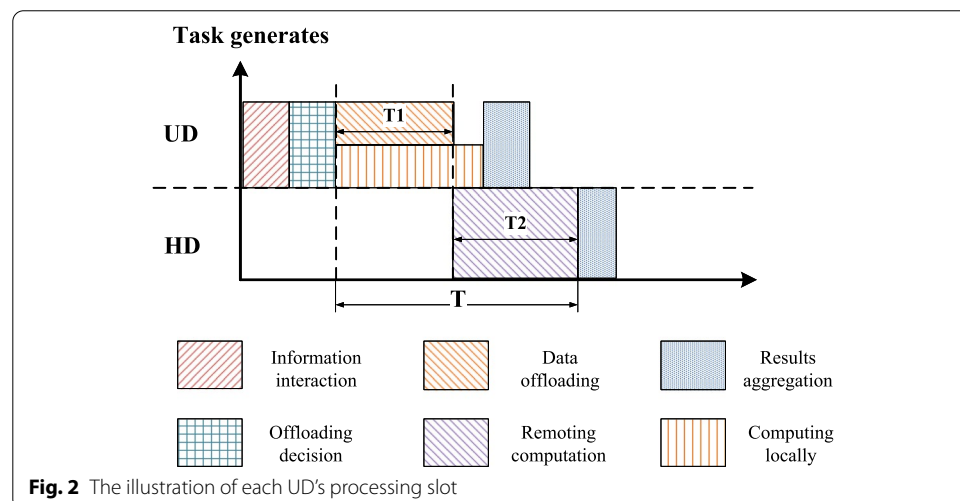
The earlier studies [18–20] investigated secure offloading for UAV-enabled MEC systems. U. A. Khan et al. [18] exploited chaotic cryptography to encrypt offloading tasks for security. Further, the study formulated an energy consumption minimization problem and solved it with heuristic algorithms. In [19], the authors considered a three-node model (i.e., a UAV, a access point, and an eavesdropper) and studied the energy-efficiency and secure offloading problem. The Access Point (AP) transmitted the noise signals to the eavesdropper for degrading the eavesdropping quality. Besides, [20] also considered the secure communication problem in the UAV-enabled MEC system. Both the legitimate UAV and idle ground users transmit jamming signals, alleviated the existence of passive eavesdropping.

Despite prior investigations on energy-efficiency and secure offloading mechanisms, there are a few differences between the current work and the afore-discussed studies. The specific expression is as follows: 1) We propose a secure channel access scheme to avoid eavesdropping attacks, as the existing anti-eavesdropping approaches cannot completely eliminate the risk of eavesdropping. 2) We study the distributed offloading scheme in UAV-enabled air-to-ground MEC Networks.

3 Method

The aim of this paper is to study a distributed energy-efficient and secure offloading scheme in air-to-ground MEC networks. This paper designs a task offloading mechanism, which considers the delay sensitivity of user tasks and the limited computing resources of mobile equipments. What's more, the energy consumption and eavesdropping rate minimization problem is formulated for system energy-constraint and physical layer security. To solve the complex non-convex problem, the two-stage offloading optimization framework is designed. The two-stage offloading scheme based on convex optimization method and learning-based algorithm achieves the sub-optimal solution with low complexity.

The effectiveness of the two-stage offloading scheme are supported by the experimental evaluation with the random generated network topology and user parameters, while the simulations have been carried out with Matlab R2017b.



4 System model and problem formulation

4.1 System model

In this paper, we consider a heterogeneous UAV-enabled air-to-ground MEC network as depicted in Fig. 1. In the MEC network, there are K ground users (Users Devices, i.e., UD), W legitimate UAV nodes (Helpers Devices, i.e., HDs) and an eavesdropper (EV). UDs have low latency and critical tasks to process and so need to seek assistance from HDs under some circumstance. To reduce tasks execution costs while fulfilling demand, UDs need to optimize their offloading decision. Notably, the computing capacity of HDs is more powerful than UDs. Hence, UDs can avail better service quality (e.g., lower delay, lower energy consumption) from HDs in the constraints of the computing capacity of HDs.

At the beginning period, the tasks that need to be processed are generated at the UDs and it has two ways to process them at this time: (1) Computing all tasks locally, (2) Selecting a HD to assist themselves. In some cases, UDs choosing to compute all tasks locally cannot complete the missions under the task delay constraint. Therefore, UDs must offload the tasks to idle HDs at this time. For simplicity, we assume that each UD selects one HD as the helper and a specific HD can assist an unlimited number of UDs under resources constraint. Notably, for multiple UDs choosing the same HD, the computing resources allocated by each UD will reduce.

Nevertheless, the UDs deciding to offload tasks will exploit a single channel to transmit the offloading data and interfere with other UDs occupying the same channel. UDs and HDs finishing the computing tasks will transmit the results of the computing tasks to the logical cluster head (pre-assigned by a specified UAV) for data aggregation.

However, during data offloading, malicious eavesdroppers can obtain the data due to the sharing of wireless channels, which leads to user safety and privacy violations. Therefore, considering security measurement as a factor is essential in making offloading decisions.

UDs and HDs sets are $\mathcal{K} = \{1, \dots, k, \dots, K\}$ and $\mathcal{W} = \{1, \dots, w, \dots, W\}$, respectively, and the set of available channel is $\mathcal{M} = \{1, \dots, m, \dots, M\}$. Besides, suppose there is a multi-antenna eavesdropper EV. EV eavesdrops on multiple channels. We define the position of legitimate UAV w as (x_w, y_w, H) , $w \in \mathcal{W}$. Similarly, the position of EV is (x_{EV}, y_{EV}, H) and the position of UD k is $(x_k, y_k, 0)$, $k \in \mathcal{K}$. Essentially, UDs can estimate the location of EV [40]. Therefore, we assume that the location of EV is known a priori. Considering the air-to-ground channel as the line-of-sight (LOS) link, the channels power gain from UD k to HD w and EV is $h_{k,w} = \frac{\beta_0}{(x_k - x_w)^2 + (y_k - y_w)^2 + H^2}$ and $h_{k,EV} = \frac{\beta_0}{(x_k - x_{EV})^2 + (y_k - y_{EV})^2 + H^2}$. Here, β_0 represents the channel power at the reference distance $d_0 = 1$. Thus, the data transmission rate between UD k and HD w is as follows:

$$R_{k,w} = B \log_2 \left(1 + \frac{p_{k,w} h_{k,w}}{N_0 + \sum_{i \in \mathcal{K} \setminus \{k\}, j \in \mathcal{W}: q_{i,j} = q_{k,w}} p_{i,j} h_{i,w}} \right). \quad (1)$$

Here B is the channel bandwidth, N_0 denotes the background noise power, and $p_{k,w}$ represents transmission power of UD k . Similarly, the eavesdropping rate (if it exists) between UD k and EV is as follows:

$$c_k = B \log_2 \left(1 + \frac{p_{k,EV} h_{k,EV}}{N_0 + \sum_{i \in \mathcal{K} \setminus \{k\}, j \in \mathcal{W}: q_{ij}=q_{k,w}} p_{i,j} h_{i,w}} \right). \quad (2)$$

Here $p_{k,EV}$ is the transmission power between UD k and eavesdropper EV. Though, the exact value of $h_{k,EV}$ is difficult to obtain. However, the probability distribution can usually be estimated [41]. Therefore, the value of $h_{k,EV}$ can be used as a priori in this paper.

The computing task of UD k ($k \in \mathcal{K}$) can be defined as $v_k = \{\gamma_k, \eta_k, \tau_k\}$, where γ_k is the number of tasks to be processed in UDs, η_k represents the number of CPU cycles per byte required for computing when UDs and HDs process tasks, τ_k is the maximum delay requirements of tasks processing. Let $\mathcal{S}_k = \{\mathcal{D}_k, \mathcal{Q}_k\}$ indicate the strategy profile of UD k , where $\mathcal{D}_k = \{\rho_k^w | w \in \mathcal{W}, 0 \leq \rho_k^w \leq 1\}$ represents the UD k 's task assignment and association with HDs. For example, $\rho_k^w = 0$ when UD k decides to process all tasks locally and $\rho_k^w = 1$ when it transmits all task data to HD w . $\mathcal{Q}_k = \{q_{k,w}\}$ is the transmission channel between UD k and HD w when computation offloading exists; note that $q_{k,w}$ represents the channel selected by UD k . Particularly, $q_{k,w} = 0$ represents that UD k computes all tasks locally. UD k 's work process has five phases, as shown in Fig. 2.

4.1.1 Global information interaction and offloading decision

Above all, all UDs and HDs begin to interact with others after the task generation. Interactive information includes task delay, the size of tasks (only UDs need to provide this information), and computing capacity (of both UDs and HDs). Further, UDs exploit an efficient task offloading mechanism to decide the sub-optimal strategy (including UDs' offloading rate, offloading object, and transmission channel) and the sub-optimal strategy should improve the energy efficiency and safety of the air-to-ground MEC system as much as possible under the constraints of the tasks delay and the computing capacity.

4.1.2 Tasks offloading

According to (1), the transmission time of task data (offloading part) of UD k is:

$$t_k^{\text{off}} = \frac{\gamma_k \rho_k^w}{R_{k,w}}. \quad (3)$$

Accordingly, the communication energy consumption of UD k is:

$$E_{k,w}^{\text{off}} = \frac{\gamma_k \rho_k^w}{R_{k,w}} \cdot p_{k,w}. \quad (4)$$

The set of UDs helped by HD w are $\mathcal{Z}_w = \{z_{w,1}, z_{w,2}, \dots, z_{w,l_w}\}$, where l_w is the number of UDs helped by HD w . We suppose that each HD equally distributes the computing resources to the assisted UDs.

4.1.3 Remoting computation

The association HD of UD k begins to implement computing tasks after task offloading. The time duration in this phase is:

$$t_k^{\text{remote}} = \frac{\gamma_k \rho_k^w \eta_k}{\alpha_w f_w}. \quad (5)$$

Here, f_w is the computing capacity of HD w (in GHz) and $\alpha_w = \frac{1}{l_w}$. Then, the overall energy consumption of w (HDs only have the computing energy consumption) is as follows:

$$E_w^c = \kappa_w \sum_{n \in \mathcal{D}_w} \gamma_n \rho_n^w \eta_n \alpha_w^2 f_w^2. \quad (6)$$

Here κ_w is a factor indicating the effective capacitance coefficient related to the hardware architecture of the computing units [42].

Local computation: Then the time for UD k to perform local computing is as follows:

$$t_k^c = \frac{\gamma_k \eta_k (1 - \rho_k^w)}{f_k}. \quad (7)$$

Here f_k denote the CPU frequency of UD k (also in GHz). Therefore, the energy consumption of UD k 's local computing is:

$$E_k^c = \kappa_k \gamma_k \eta_k f_k^2 (1 - \rho_k^w). \quad (8)$$

Here κ_k is a constant coefficient which reflects the energy consumption of CPU when processing data tasks (same as κ_w above). Finally, for the tasks generated by UD k , its whole execution time is:

$$T_k^{\text{total}} = \max \left\{ t_k^{\text{off}} + t_k^{\text{remote}}, t_k^c \right\}. \quad (9)$$

The overall energy consumption of UD k (including computing part and communication part) is:

$$E_k = E_k^c + E_k^{\text{off}}, \quad (10)$$

while the whole energy consumption of HD w (only the computing part) is:

$$E_w = E_w^c. \quad (11)$$

4.1.4 Results aggregation

After performing all tasks, both UD_s and HD_s will upload results to the logical cluster head (a pre-specified role, played by a legitimate UAV) for data aggregation. The cluster head will collect the data analysis results or forwarding them to other nodes needing information. Most of the existing research works related to MEC usually ignore the size of the data. Similarly, we do not consider the data size of the results in this study and thus, ignoring the time spent in this phase.

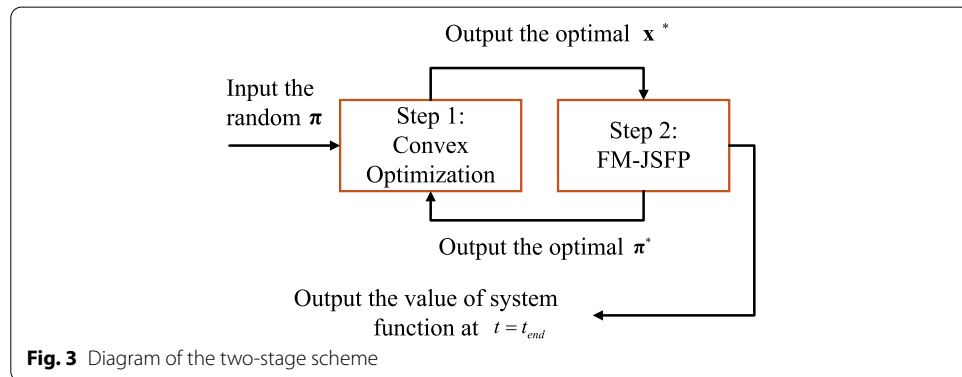
In this paper, we consider a quasi-static scenario in which all mobile devices states have no change during an offloading time block (e.g., within a few hundred milliseconds). In other words, the location relationship between mobile devices is almost unchanged, and the UD_s does not need to process the newly arrived task data.

4.2 Problem formulation

In this section, the objective is minimizing the weighted sum of system eavesdropping rate and energy consumption under multiple constraints. The optimization problem is as follows:

$$\begin{aligned}
 (\mathcal{P}0) : \quad & \min_{\mathcal{D}_k, \mathcal{Q}_k, \forall k \in \mathcal{K}} \varepsilon_1 \left(\sum_{k \in \mathcal{K}} C_k \right) + \varepsilon_2 \left(\sum_{k \in \mathcal{K}} E_k + \sum_{w \in \mathcal{W}} E_w \right) \\
 \text{s.t.} \quad & t_k^{\text{remote}} + t_k^{\text{off}} \leq T_k, \forall k \in \mathcal{K}, \quad \mathbf{C1} \\
 & t_k^c \leq T_k, \forall k \in \mathcal{K}, \quad \mathbf{C2} \\
 & 0 \leq \rho_k^w \leq 1, \forall k \in \mathcal{K}, \quad \mathbf{C3}.
 \end{aligned} \tag{12}$$

Here, the E_k and E_w indicate the respective energy consumption of UD k and HD w and ε_1 and ε_2 are weighting factors. C_k represents the eavesdropping rate from UD k to the eavesdropper, and $C_k = c_k$. The constraint **C1** represents that the remote computing and offloading time of UD k should less than T_k (corresponds to the constraints of task delay). The constraint **C2** indicates that the time of local computing of each UD cannot exceed T_k . Besides, the constraint **C3** shows that the range for tasks offloading rate of UD k should be within $[0, 1]$. The system energy consumption reduces when UD_s offload the data to HD_s, but it increases the risk of eavesdropping. Therefore, it is a trade-off between low energy consumption offloading and safe communication. UD_s optimize the optimization variables (i.e., offloading rate, offloading object, and offloading channel) to minimize the objective function.



5 Secure and green offloading scheme

5.1 Problem reformulation

The above optimization problem is a mixed-integer nonlinear programming (MINLP) problem and the optimization variables are coupled with each other. Further, the closed-form solution of this optimization problem is difficult to obtain. Hence, we reformulate the optimization problem ($\mathcal{P}0$) to task assignment and UD's association sub-problem, and adding auxiliary variables. The steps are as follows.

First, design a decision matrix $\pi = \{\pi_{ij}\}, \pi_{ij} \in \mathbb{R}^{K \times W}$, where $\pi_{ij} \in \{0, 1, 2, \dots, M\}$ indicates tasks offloading object and the transmission channel. For example, π_{ij} represent that UD i chooses to offload the tasks to HD j by channel π_{ij} (especially $\pi_{i,:} = 0$ if UD i executes computing locally). Notably, the number of non-zero elements in every row of π is 1 (i.e., each UD can only find one HD to assist in the computing) and define a function $\delta(\cdot)$ for redescribing of optimization goals:

$$\delta(\pi_{ij}) = \begin{cases} 0, & \pi_{ij} = 0 \\ 1, & \pi_{ij} = 1. \end{cases} \quad (13)$$

A offloading rate vector $\mathbf{x} = \{x_1, x_2, \dots, x_K\}$ represents each UD's offloading rate, and the optimal problem is:

$$\begin{aligned} \mathcal{P}1 : \min & \varepsilon_1 C(\pi) + \varepsilon_2 E(\pi, \mathbf{x}) \\ \text{s.t.} & \text{C1} - \text{C3}. \end{aligned} \quad (14)$$

where $C(\pi) = \sum_{k \in \mathcal{K}} C_k$. In the following section, we refer to the optimization objective as a "system function". The whole energy consumption of the system (including all UD's and HD's) $E(\pi, \mathbf{x})$ is as follows:

$$\begin{aligned} E(\pi, \mathbf{x}) = & \sum_{k \in \mathcal{K}} \gamma_k \left(\frac{p_{k,w}}{R_{k,w}} - \kappa_k \eta_k f_k^2 \right) x_k \\ & + \sum_{k \in \mathcal{K}} \kappa_k \gamma_k \eta_k f_k^2 + \sum_{k \in \mathcal{K}} \sum_{m \in \mathcal{W}} \delta(\pi_{k,m}) \gamma_k \eta_k \alpha_m^2 f_m^2 \kappa_m x_k \end{aligned} \quad (15)$$

To more clearly describe the problem ($\mathcal{P}1$), we convert the form of ($\mathcal{P}1$) to ($\mathcal{P}1'$):

$$\begin{aligned} \mathcal{P}1' : \min & \varepsilon_1 \sum_{k \in \mathcal{K}} C_k + \varepsilon_2 \sum_{k \in \mathcal{K}} (\psi_k + \zeta_k) x_k + \varepsilon_2 \phi \\ \text{s.t.} & \text{C1} - \text{C3}. \end{aligned} \quad (16)$$

Here constant $\psi_k = \gamma_k \left(\frac{p_{k,w}}{R_{k,w}} - \kappa_k \eta_k f_k^2 \right)$, $\zeta_k = \sum_{m \in \mathcal{W}} \delta(\pi_{k,m}) \gamma_k \eta_k \alpha_m^2 f_m^2 \kappa_m$, and $\phi = \sum_{k \in \mathcal{K}} \kappa_k \gamma_k \eta_k f_k^2$. Notably, for a fixed π , these variables have fixed values.

5.2 A distributed two-stage offloading scheme

At the first stage, the ratio between $T_{1,k}$ and $T_{2,k}$ ($T_{1,k} + T_{2,k} = T_k$) is fixed for simplicity. The function of $T_{1,k}$ and $T_{2,k}$ is to limit the maximum value of t_k^{off} and t_k^{remote} (i.e., $t_k^{\text{off}} \leq T_{1,k}$, $t_k^{\text{remote}} \leq T_{2,k}$). Splitting T_k was to prevent UD's from making unreasonable decisions. For instance, larger values of t_k^{off} represents that UD's offload numerous tasks to HD's, which can obstruct the completion of the offloading tasks by HD's within the

Table 1 Simulation parameters setting

| Parameters | Value |
|--|--|
| Transmission power p_k | 400 mW |
| Background noise N_0 | − 100 dBm |
| Computing capacity of UDs (GHz) | $\mathcal{U}[1.5, 2]$ |
| Computing capacity of HDs (GHz) | $\mathcal{U}[4, 4.5]$ |
| Capacitance coefficients κ_k, κ_w | $1 \times 10^{-27}, 3 \times 10^{-28}$ |
| The amount of UDs K | 5 |
| Input data (Mb) | $\mathcal{U}[1.4, 1.6]$ |
| η_k | $\mathcal{U}[800, 1000]$ |
| Task processing delay (ms) | $\mathcal{U}[500, 700]$ |
| Channel bandwidth B | 15 MHz |
| The number of channels M | 3 |
| Height of UAVs (m) | 100 |
| Weighting factor ε_1 and ε_2 | $1 \times 10^{-6}, 2$ |
| The channel power gain β_0 (dB) | $\mathcal{U}[-22, -18]$ |
| λ, θ | 0.5, 0.02 |

maximum delay constraint. Thus, setting a reasonable ratio between $T_{1,k}$ and $T_{2,k}$ is imperative. So, constraints for C1, C2 are as follows:

$$\begin{aligned} x_k &\leq \sigma_k, \mathbf{C1}' \\ x_k &\leq \chi_k, \mathbf{C2}' \\ x_k &\geq \varsigma_k, \mathbf{C3}' \end{aligned} \quad (17)$$

Here constant $\sigma_k = \frac{T_{2,k}\alpha_w(k)f_{w(k)}}{\gamma_k\eta_k}$, $\chi_k = \frac{R_{k,w}T_{1,k}}{\gamma_k}$, and $\varsigma_k = 1 - \frac{(T_{1,k}+T_{2,k})f_k}{\gamma_k\eta_k}$. $w(k)$ represents the particular HD that assisted UD k .

First, a initial decision matrix π_0 is randomly produced. After fixing the decision π_0 , the system function and offloading rate \mathbf{x} become linear. Analyzing the monotonicity of the system function (using convex optimization theory) provides the following optimal offloading ratio x_k^* :

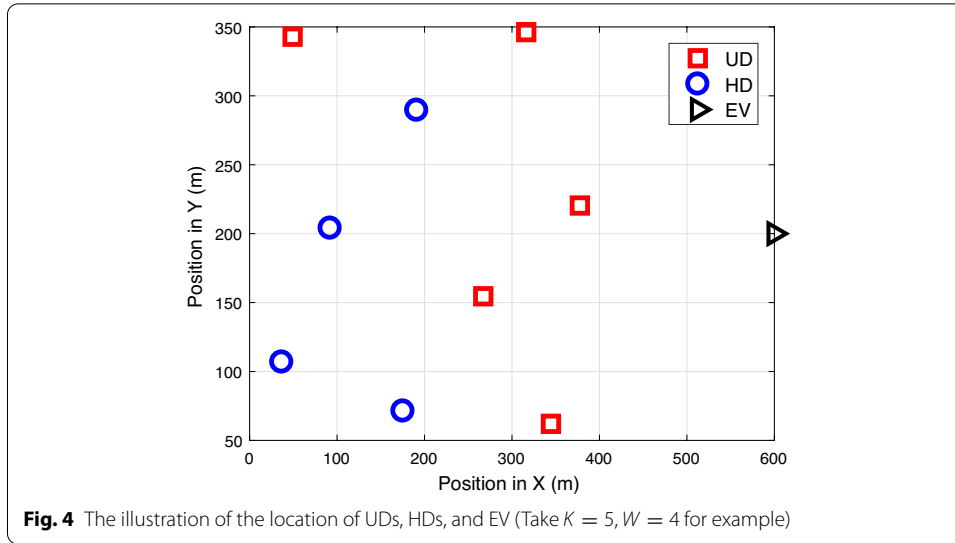
$$x_k^* = \begin{cases} \max\{\varsigma_k, 0\}, & \text{if } \exists \pi_{k,:} \neq 0 \text{ and } \psi_k + \zeta_k \geq 0 \\ \min\{\sigma_k, \chi_k, 1\}, & \text{if } \exists \pi_{k,:} \neq 0 \text{ and } \psi_k + \zeta_k < 0 \\ 0, & \text{if } \forall \pi_{k,:} = 0. \end{cases} \quad (18)$$

At the second stage, based on the x_k^* of the first stage, the optimization objective of the system is expressed as:

$$\min \varepsilon_1 C(\pi) + \varepsilon_2 E(\pi, \mathbf{x}^*). \quad (19)$$

The above expression depicts that its value depends on the matrix variable π only.

Next, we introduce a learning-based distributed algorithm for joint assignment of computing and channel resources mainly based on the fading memory joint strategy fictitious play with inertia (FM-JSFP). FM-JSFP is an efficient distributed learning method for the offloading decision problem as it has computational processability in large networks. Further, it can exploit historical information to make decisions. The discount factor λ represents the crucial degree of historical information. After the fixing of first decision x_k^* , the strategy set in the t -th slot of UD

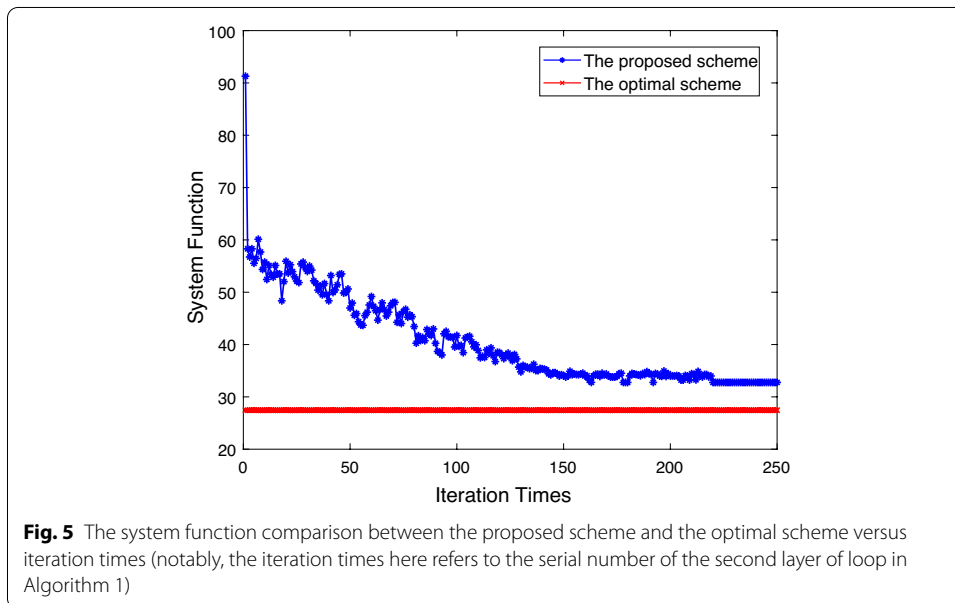


k is expressed as $\mathcal{S}_k = \{(w_k, m_k) | w_k \in \mathcal{W}, m_k \in \{1, 2, \dots, M\}, \forall k \in \mathcal{K}\}$, which corresponds to the strategy matrix π . In this distributed algorithm, the utility of each UD is:

$$u(S_k(t), S_{-k}(t)) = \varepsilon_1 C_k + \varepsilon_2 (\psi_k + \zeta_k) x_k + \varepsilon_2 \kappa_k \gamma_k \eta_k f_k^2. \quad (20)$$

where $S_k(t)$ indicates the action in the t -th slot of UD k and $S_{-k}(t)$ represents the action except for UD k . Notably, the sum of the utility of all UD is the “system function”.

To summarize, Algorithm 1 enlists the specific steps, and Fig. 3 shows the graphical illustration of the distributed two-stage scheme.



Algorithm 1 Distributed Learning-based Offloading Algorithm (DLOA)

Initialization Setting $n = 1, m = 1$ and each UD selects a random action $S_k(t) \in \mathcal{S}_k$. At the same time, each UD can interact with other UDs.

Loop for $n = 1, 2, 3, \dots, N_{max}$

Based on the initial decision (i.e., π_0), making a optimal task assignment decision \mathbf{x}^* . Then, recording the value of the system function (called $F1$).

Do While $m \leq I_{max}$

{ **Loop for** $t = 1, 2, 3, \dots, T_{end}$

Belief estimate: each UD k trials all available actions, and obtains the utility $V_k^{S_k(1)}(1) \in R$ for each $S_k(1) \in \mathcal{S}_k$, where $V_k^{S_k(1)}(1) = u(S_k(1), S_{-k}(1))$. $V_k^{S_k(1)}(1)$ is the initial belief of UD k about the reward it would achieve for executing action S_k in the following time step.

Action selection: At slot t , each UD k executes the follow strategy:

• $S_k(t) \in \arg \max_{S_k(t) \in \mathcal{S}_k} V_k^{S_k(t)}(t)$ with probability $1 - \theta$,

• $S_k(t) = S_k(t-1)$ with probability θ .

Belief propagation: Each UD k updates its beliefs as

$$V_k^{S_k}(t+1) = (1 - \lambda) V_k^{S_k}(t) + \lambda U_k(S_k(t), S_{-k}(t)), \forall S_k \in \mathcal{S}_k,$$

where $\lambda \in (0, 1]$ is the discount factor and $U_k(S_k(t), S_{-k}(t)) = \Delta_k \cdot u_k(S_k(t), S_{-k}(t))$. Note that when UD k 's action satisfies the constraint, the value of Δ_k is -1 , otherwise it is -1.4 (as punishment).

End loop

Recording the sum utility of all UDs in the T_{end} slot (i.e., the value of $\sum_{k \in \mathcal{K}} u_k(S_k(T_{end}), S_{-k}(T_{end}))$) and its corresponding π^* . Let $m = m + 1$. If the value of system function less than $F1$, skipping to the loop $n + 1$ and outputting the decision π^* . }

Outputting the decision π in previous loop $n - 1$.

End loop when $n = N_{max}$.

According to the outputting decision matrix π , calculating the value of system function.

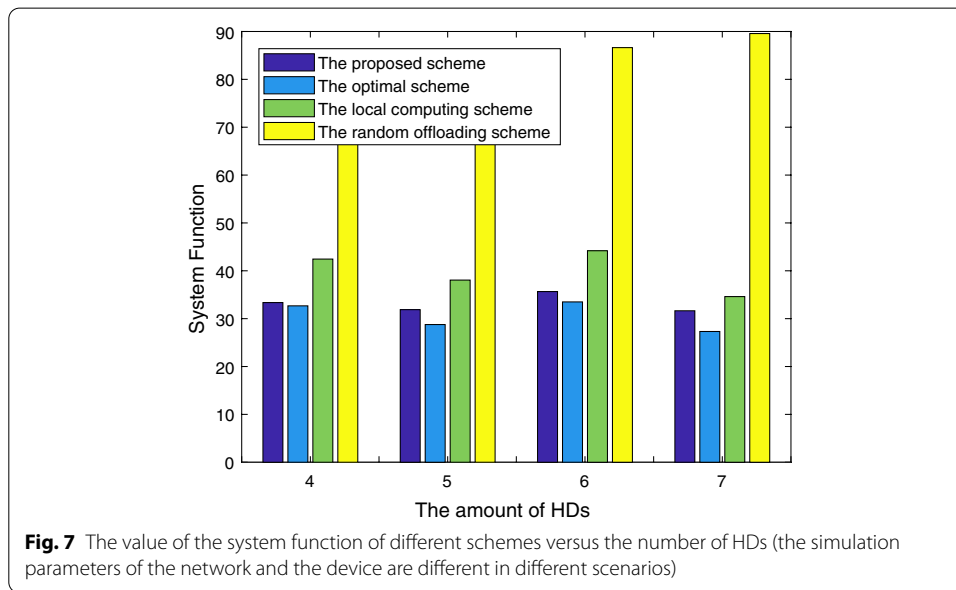
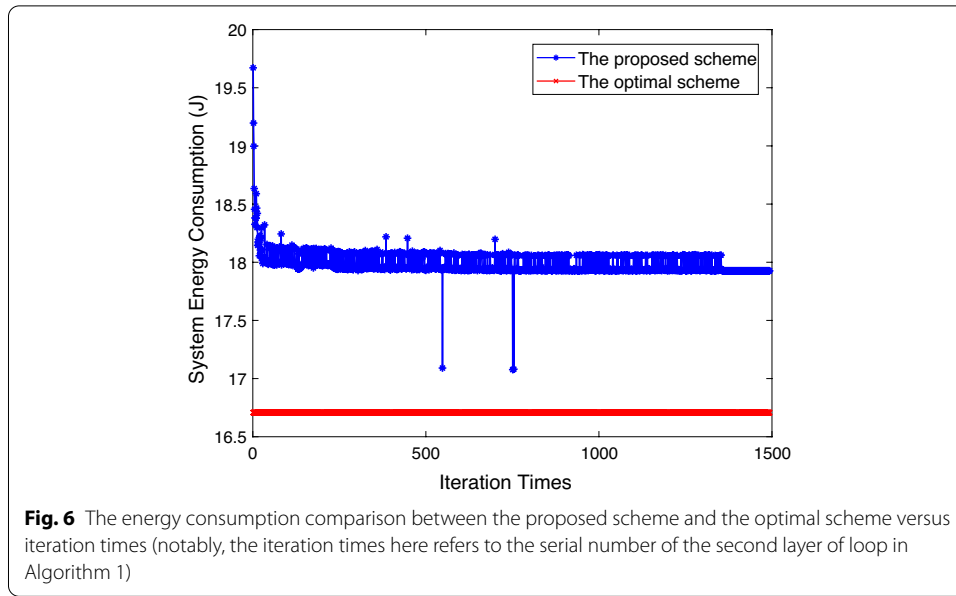
For further clarification, we describe the rules for setting the penalty value (i.e., Δ_k) in detail. We add the penalty value to the utility for the decisions generated (i.e., π) by Algorithm 1 make the next iteration decision (i.e., \mathbf{x} at the next moment) infeasible. Specifically, by introducing penalty value Δ_k into utility, UDs tend to choose actions without penalty value in the process of iteration. Meanwhile, when decision π without penalty value is fixed, there is a feasible solution for decision variable \mathbf{x} in the next iteration. In other words, the penalty value eliminates the decision (corresponding to matrix π) that does not satisfy the constraints.

5.3 Algorithm complexity analysis

First, in each slot n , each UD selects a optimal offloading rate \mathbf{x} , and the computation complexity is $\mathcal{O}(C_1K)$, where C_1K is a small constant. Then, each UD runs the FM-JSFP algorithm to decide optimal π based on \mathbf{x} above. In the worst case, the FM-JSFP algorithm needs to run I_{max} times. When the FM-JSFP algorithm runs, UDs select actions and the computation complexity is $\mathcal{O}(C_2K)$, where C_2K is a small constant. Then, UDs compute their utilities and update belief. The corresponding computation complexity is $\mathcal{O}(C_3K)$, where C_3K is a small constant. So the computation complexity of the FM-JSFP algorithm is $T_{end}(\mathcal{O}(C_2K) + \mathcal{O}(C_3K))$, where T_{end} is the maximum iteration times of the FM-JSFP algorithm. In summary, the time complexity of Algorithm 1 in the worst case is:

$$N_{max}(\mathcal{O}(C_1K) + I_{max}T_{end}(\mathcal{O}(C_2K) + \mathcal{O}(C_3K))), \quad (21)$$

where N_{max} is the maximum iteration times of the outer loop.



6 Simulation results and discussion

6.1 Simulation parameters

In this subsection, we illustrate the main simulation parameter values with Table 1. Notably, $X \sim \mathcal{U}[b, c]$ indicates that X obeys the uniform distribution within $[b, c]$. Besides, the computing capacities setting of UDs are weaker than HDs, which is similar to [43].

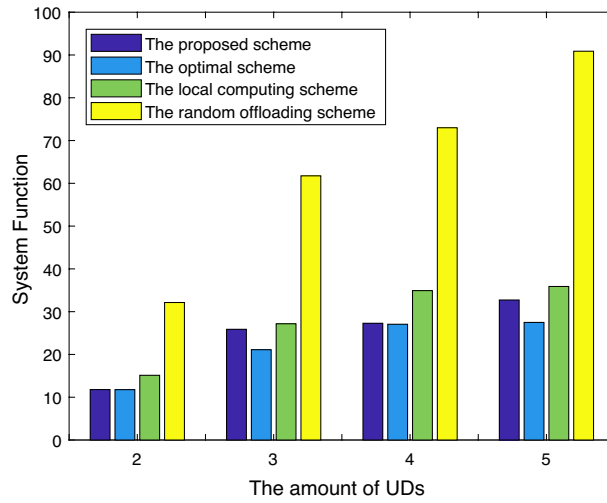


Fig. 8 The value of the system function of the proposed scheme versus the number of UDs (the simulation parameters of the network and the device are different in different scenarios)

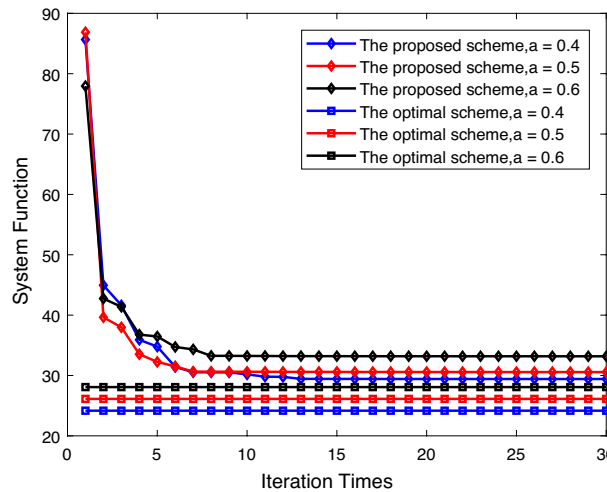


Fig. 9 The value of the system function of the proposed scheme and the optimal scheme versus different parameter a (a refers to the ratio of $T_{1,k}$ to $T_{2,k}$ mentioned above) and iteration times here refers to the outer loop in Algorithm 1

6.2 Network topology

Figure 4 demonstrates a random topology of $400m$ -by- $400m$ in air-to-ground MEC networks (the location of the eavesdropping UAV is far away). There are five UDs with different mission requirements which probably offload data to HDs and four idle HDs which can assist UDs in computing tasks, i.e., $\mathcal{K}=\{1, 2, 3, 4, 5\}$ and $\mathcal{W}=\{1, 2, 3, 4\}$.

6.3 Algorithm convergence and performance comparison

Multiple benchmark schemes mentioned in the simulation results are as follows:

- *The Optimal Scheme* It is an exhaustive method, which achieves the theoretically optimal solution has high algorithm complexity. With the increases in decision space, dimensionality explosion can occur. Further, the time complexity is $\mathcal{O}((M \cdot W)^K)$.
- *The Local Computing Scheme* All UD's perform computing tasks locally without considering task delay constraints.
- *The Random Offloading Scheme* UD's randomly select the offloading object and the transmission channel. The system function value for this scheme is the average of multiple actions (with the simulation number set to 1000).

Figure 5 presents the system function values with the proposed scheme and optimal schemes versus the iteration times. Further, we average the multiple simulation results for the algorithm provides the curve shown in the figure. It can be seen that the algorithm reaches convergence after several iterations. Besides, the performance gap between proposed and optimal scheme is small (reach about 10%).

Figure 6 presents the energy consumption values with the proposed scheme and optimal schemes versus the iteration times (let $\varepsilon_1 = 0$ and $\varepsilon_2 = 1$). Similarly, to eliminate the randomness, we run the algorithm many times (the order of 100) and get the average value. It can be seen that the algorithm reaches convergence after several hundreds of iterations. What's more, the energy consumption of the proposed scheme is similar to that of the optimal scheme (only about 7% more). On the other hand, Fig. 6 also shows that the performance of the proposed scheme in a single target is better with low computational complexity.

Figure 7 demonstrates the value of the system function of different schemes under different number of HDs. In various scenarios, the performance of the proposed offloading scheme is close to the optimal offloading scheme. As the number of HDs changes, the reason why the optimal value of the system function is almost unchanged is that UD's workload and channel conditions have hardly changed. In terms of the value of the system function, the proposed scheme is better than that of the local computing scheme and the random offloading scheme.

Figure 8 presents the performance of the proposed scheme under different number of UD's. Notably, the number of HDs and available channel is 4 and 3, respectively. It shows that as the number of UD's grows (represent the increase of task load), the value of the system function continues to rise. Besides, the gap between the performance of the proposed scheme and the optimal scheme is small (the worst-case performance gap is 18%). Although the gap between the local computing scheme and the proposed scheme is not large, the local computing scheme cannot generally meet UD's task delay requirements.

Figure 9 demonstrates that the convergence value of the proposed offloading scheme is close to the optimal offloading scheme for different parameters a . And under different parameter settings, the convergence rate is about the same. Notably, parameter a can affect the system function value and it is application dependent. Thus, the decision of the parameter a need to be inspected in practical tests.

7 Conclusion

This paper investigated security and green offloading problem in air-to-ground MEC networks. To solve the Mixed-integer Nonlinear Programming (MINLP) problem, the optimization problem was decomposed into two sub-optimization problems and the optimization variables were decoupled. Further, we proposed a distributed two-stage offloading scheme, which considered the requirements of user devices (UDs) for heterogeneous tasks, the heterogeneous computing capacity of helper devices (HDs), and the presence of eavesdroppers (EV). The proposed scheme based on the convex optimization and learning approach can successfully realize the secure and energy-efficient offloading for the system. Simulation results show that the proposed offloading scheme achieves better performance than benchmark schemes.

Abbreviations

MEC: mobile edge computing; MINLP: mixed-integer nonlinear programming; UAVs: unmanned aerial vehicles; BSs: base stations; APs: access points; D2D: device-to-device; FM-JSFP: fading memory joint strategy fictitious play with inertia; IoTD: internet of things devices; MTs: mobile terminals; UEs: user equipments; TDD: time division duplexing; RL: reinforcement learning; UD: users devices; HDs: helpers devices.

Acknowledgements

Not applicable.

Authors' contributions

WL contributed in the system model and performed the simulation tests. Besides, WL was a major contributor in writing the manuscript. YX proposed improvements and corrections to the system model and problem formulation. DW assisted in the design of a method of coping with complex optimization problems and provided a lot of meaningful references. HW and XZ assisted simulation experiment. XC participated in the experiment and analysis. All authors read and approved the final manuscript.

Funding

This work was supported by the National Natural Science Foundation of China under Grant Nos. 61901517, 61771488, 62001514, and in part by the National Postdoctoral Program for Innovative Talents under Grant BX2021371.

Availability of data and materials

Data and materials sharing not applicable to this paper as no datasets were generated or analyzed during the current study.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

The manuscript does not contain any individual person's data in any form (including individual details, images or videos) and therefore the consent to publish is not applicable to this article.

Competing interests

The authors declare that they have no competing interests.

Received: 20 May 2021 Accepted: 9 August 2021

Published online: 17 August 2021

References

1. Y. Mao, C. You, J. Zhang, K. Huang, K.B. Letaief, A survey on mobile edge computing: the communication perspective. *IEEE Commun. Surv. Tutor.* **19**, 2322–2358 (2017)
2. Y. Zhao, W. Wang, Y. Li, C. Colman Meixner, M. Tornatore, J. Zhang, Edge computing and networking: a survey on infrastructures and applications. *IEEE Access* **7**, 101213–101230 (2019)
3. R. Amorim, H. Nguyen, P. Mogensen, I.Z. Kovacs, J. Wigard, T.B. Sorensen, Radio channel modeling for UAV communication over cellular networks. *IEEE Wirel. Commun. Lett.* **6**, 514–517 (2017)
4. A. Al-Hourani, K. Gomez, Modeling cellular-to-UAV path-loss for suburban environments. *IEEE Wirel. Commun. Lett.* **7**, 82–85 (2018)
5. Z. Yang, C. Pan, K. Wang, M. Shikh-Bahaei, Energy efficient resource allocation in UAV-enabled mobile edge computing networks. *IEEE Trans. Wirel. Commun.* **18**, 4576–4589 (2019)

6. Shakhathreh et al., Unmanned aerial vehicles (UAVs): a survey on civil applications and key research challenges. *IEEE Access* **7**, 48572–48634 (2019)
7. I. Bekmezci, O. Sahingoz, a Temel, Flying ad-hoc networks (FANETs): a survey. *Ad Hoc Netw.* **11**, 1254–1270 (2013)
8. J. Xu, L. Duan, R. Zhang, Surveillance and intervention of infrastructure-free mobile communications: a new wireless security paradigm. *IEEE Wirel. Commun.* **24**, 152–159 (2017)
9. H. Lu, H. Zhang, H. Dai, W. Wu, B. Wang, Proactive eavesdropping in UAV-aided suspicious communication systems. *IEEE Trans. Veh. Technol.* **68**, 1993–1997 (2019)
10. J. Ye, C. Zhang, H. Lei, G. Pan, Z. Ding, Secure UAV-to-UAV systems with spatially random UAVs. *IEEE Wirel. Commun. Lett.* **8**, 564–567 (2019)
11. S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, W. Wang, A survey on mobile edge networks: convergence of computing, caching and communications. *IEEE Access* **5**, 6757–6779 (2017)
12. M. Mehrabi, D. You, V. Latzko, H. Salah, M. Reisslein, F.H.P. Fitzek, Device-enhanced MEC: multi-access edge computing (MEC) aided by end device computation and caching—a survey. *IEEE Trans. Veh. Technol.* **7**, 166079–166108 (2019)
13. Y. Du, K. Yang, K. Wang, G. Zhang, Y. Zhao, D. Chen, Joint resources and workflow scheduling in UAV-enabled wirelessly-powered MEC for IoT systems. *IEEE Trans. Veh. Technol.* **68**, 10187–10200 (2019)
14. X. Xu et al., Secure service offloading for internet of vehicles in SDN-enabled mobile edge computing. *IEEE Trans. Intell. Transp. Syst.* **22**, 3720–3729 (2020)
15. H. Yang et al., Secure resource allocation in mobile edge computing systems, in *2019 IEEE Global Communications Conference (GLOBECOM)* (2019), pp. 1–6
16. B. Li, T. Chen, X. Wang, G.B. Giannakis, Secure edge computing in IoT via online learning, in *2018 52nd Asilomar Conference on Signals, Systems, and Computers* (2018), pp. 2149–2153
17. B. Li, T. Chen, G.B. Giannakis, Secure mobile edge computing in IoT via collaborative online learning. *IEEE Trans. Signal Process.* **67**, 5922–5935 (2019)
18. U. A. Khan, W. Khalid, S. Saifullah, Energy efficient resource allocation and computation offloading strategy in a UAV-enabled secure edge-cloud computing system, in *2020 IEEE International Conference on Smart Internet of Things (SmartIoT)* (2020), pp. 58–63
19. T. Bai, J. Wang, Y. Ren, L. Hanzo, Energy-efficient computation offloading for secure UAV-edge-computing systems. *IEEE Trans. Veh. Technol.* **68**, 6074–6087 (2019)
20. Y. Zhou et al., Secure communications for UAV-enabled mobile edge computing systems. *IEEE Trans. Commun.* **68**, 376–388 (2020)
21. S. Jeong, O. Simeone, J. Kang, Mobile edge computing via a UAV-mounted cloudlet: optimization of bit allocation and path planning. *IEEE Trans. Veh. Technol.* **67**, 2049–2063 (2018)
22. C. Zhan, H. Hu, X. Sui, Z. Liu, D. Niyato, Completion time and energy optimization in the UAV-enabled mobile-edge computing system. *IEEE IoT J.* **7**, 7808–7822 (2020)
23. H. Mei, K. Yang, Q. Liu, K. Wang, Joint trajectory-resource optimization in UAV-enabled edge-cloud system with virtualized mobile clone. *IEEE IoT J.* **7**, 5906–5921 (2020)
24. Y. Liu, K. Xiong, Q. Ni, P. Fan, K.B. Letaief, UAV-assisted wireless powered cooperative mobile edge computing: joint offloading, CPU control, and trajectory optimization. *IEEE IoT J.* **7**, 2777–2790 (2020)
25. N. Qi, N.I. Miridakis, M. Xiao et al., Traffic-aware two-stage queueing communication networks: queue analysis and energy saving. *IEEE Trans. Commun.* **68**, 4919–4932 (2020)
26. N. Qi, M. Xiao, T.A. Tsiftsis et al., Efficient coded cooperative networks with energy harvesting and transferring. *IEEE Trans. Wirel. Commun.* **16**, 6335–6349 (2017)
27. Y. Xu et al., Convert harm into benefit: a coordination-learning based dynamic spectrum anti-jamming approach. *IEEE Trans. Veh. Technol.* **69**, 13018–13032 (2020)
28. Y. Xu, A. Anpalagan, Q. Wu, L. Shen, Z. Gao, J. Wang, Decision-theoretic distributed channel selection for opportunistic spectrum access: strategies, challenges and solutions. *IEEE Commun. Surv. Tutor.* **15**, 1689–1713 (2013)
29. Y. Xu, J. Wang, Q. Wu, J. Zheng, L. Shen, A. Anpalagan, Dynamic spectrum access in time-varying environment: distributed learning beyond expectation optimization. *IEEE Trans. Commun.* **65**, 5305–5318 (2017)
30. N. Qi, W. Wang, M. Xiao et al., A learning-based spectrum access Stackelberg game: friendly jammer-assisted communication confrontation. *IEEE Trans. Veh. Technol.* **70**, 700–713 (2021)
31. Y. Xu, J. Wang, Q. Wu et al., Opportunistic spectrum access in cognitive radio networks: global optimization using local interaction games. *IEEE J. Sel. Top. Signal Process.* **6**, 180–194 (2011)
32. S.P. Boyd, L. Vandenberghe, *Convex Optim.* (Cambridge University Press, London, 2004)
33. J.R. Marden, G. Arslan, J.S. Shamma, Joint strategy fictitious play with inertia for potential games. *IEEE Trans. Autom. Control* **54**, 208–220 (2009)
34. X. Diao, J. Zheng, Y. Cai, Y. Wu, A. Anpalagan, Fair data allocation and trajectory optimization for UAV-assisted mobile edge computing. *IEEE Commun. Lett.* **23**, 2357–2361 (2019)
35. X. Diao, J. Zheng, Y. Wu, Y. Cai, A. Anpalagan, Joint trajectory design, task data, and computing resource allocations for NOMA-based and UAV-assisted mobile edge computing. *IEEE Access* **7**, 117448–117459 (2019)
36. Y. Wang, Z.-Y. Ru, K. Wang, P.-Q. Huang, Joint deployment and task scheduling optimization for large-scale mobile users in multi-UAV-enabled mobile edge computing. *IEEE Trans. Cybern.* **50**, 3984–3997 (2020)
37. H. Xiao, Z. Hu, K. Yang, Y. Du, D. Chen, An energy-aware joint routing and task allocation algorithm in MEC systems assisted by multiple UAVs, in *2020 International Wireless Communications and Mobile Computing (IWCMC): 2020* (2020), pp. 1654–1659
38. L. Wang, et al., RL-based user association and resource allocation for multi-UAV enabled MEC, in *2019 15th International Wireless Communications and Mobile Computing Conference (IWCMC), Tangier, Morocco* (2019), pp. 741–746
39. W. Liu, Y. Xu, N. Qi, K. Yao, Y. Zhang, W. He, Joint computation offloading and resource allocation in UAV swarms with multi-access edge computing, in *2020 International Conference on Wireless Communications and Signal Processing (WCSP), Nanjing* (2020), pp. 280–285

40. X. Wei, Q. Wang, T. Wang, J. Fan, Jammer localization in multi-hop wireless network: a comprehensive survey. *IEEE Commun. Surv. Tutor.* **19**, 765–799 (2017)
41. L. Jia, F. Yao, Y. Sun, Y. Niu, Y. Zhu, Bayesian Stackelberg game for antijamming transmission with incomplete information. *IEEE Commun. Lett.* **20**, 1991–1994 (2016)
42. H. Xing, L. Liu, J. Xu, A. Nallanathan, Joint task assignment and resource allocation for D2D-enabled mobile-edge computing. *IEEE Trans. Commun.* **67**, 4193–4207 (2019)
43. G. Hu, Y. Jia, Z. Chen, Multi-user computation offloading with D2D for mobile edge computing, in *2018 IEEE Global Communications Conference (GLOBECOM)* (2018)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
