EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Collaborative offloading for UAV-enabled time-sensitive MEC networks

Wen-Tao Li, Mingxiong Zhao*, Yu-Hui Wu, Jun-Jie Yu, Ling-Yan Bao, Huan Yang and Di Liu

*Correspondence:
jimmyzmx@gmail.com
National Pilot School
of Software, Yunnan
University, Kunming 650091,
China

Abstract

Recently, unmanned aerial vehicle (UAV) acts as the aerial mobile edge computing (MEC) node to help the battery-limited Internet of Things (IoT) devices relieve burdens from computation and data collection, and prolong the lifetime of operating. However, IoT devices can ONLY ask UAV for either computing or caching help, and collaborative offloading services of UAV are rarely mentioned in the literature. Moreover, IoT device has multiple mutually independent tasks, which make collaborative offloading policy design even more challenging. Therefore, we investigate a UAV-enabled MEC networks with the consideration of multiple tasks either for computing or caching. Taking the quality of experience (QoE) requirement of time-sensitive tasks into consideration, we aim to minimize the total energy consumption of IoT devices by jointly optimizing trajectory, communication and computing resource allocation at UAV, and task offloading decision at IoT devices. Since this problem has highly non-convex objective function and constraints, we first decompose the original problem into three subproblems named as trajectory optimization ($\mathbf{P_T}$), resource allocation at UAV ($\mathbf{P_R}$) and offloading decisions at IoT devices ($\mathbf{P_O}$) and then propose an iterative algorithm based on block coordinate descent method to cope with them in a sequence. Numerical results demonstrate that collaborative offloading can effectively reduce IoT devices' energy consumption while meeting different kinds of offloading services, and satisfy the QoE requirement of time-sensitive tasks at IoT devices.

Keywords: UAV, MEC, QoE, Collaborative offloading, Trajectory, Time-sensitivity

1 Introduction

With the rapid development of internet of things (IoT), numerous novel applications and services, such as augmented/virtual reality (AR/VR), face recognition and e-health, have emerged and interacted with each other [1], which start demanding more on computing capability, network throughput and latency for better quality of service (QoS) [2, 3]. However, the heavy computation workload and storage demand lay a burden on the resource-limited IoT devices. To find a way out of this dilemma, mobile edge computing (MEC) has been envisioned as the paradigm by placing servers with rich services in proximity to IoT devices, which can remarkably improve the quality of experience (QoE) of IoT devices and help to reduce the energy dissipation [4–8].

In general, the MEC server deployment is fixed, restraining it from moving closer to IoT devices in areas without communication coverage or in special cases when it is difficult to deploy MEC servers. Hence, UAV is extensively employed as an aerial MEC node for service coverage in surveillance, data collection, disaster relief, and public safety due to its excellent flexibility and mobility [9]. Although UAV-assisted MEC provides IoT devices with remote resources, it still faces the challenge of communication and computation design due to the limited embedded battery of UAV and IoT devices, and several prior related works have been done to minimize the energy consumption of ground or sensor nodes (GNs/SNs) [10–12]. To be specific, the total UAV energy consumption while satisfying the communication throughput requirement of GN was investigated in [10], and the authors in [11] employed UAV for minimizing the maximum energy consumption of SNs. In addition, the energy consumption tradeoff between UAV and GNs had been investigated in [12], where authors attempted to reduce the overall weighted energy consumption between SNs and UAV, while ensuring the required amount of data collection from each SN just as [11] investigated.

However, offloading energy-consuming workloads to UAV also invokes extra latency which significantly affects the QoE of IoT devices, and thus cannot be ignored in the system design. Recently, some inspiring related works have considered the QoE requirement of time-sensitive tasks of IoT devices [13–17]. Specifically, the energy consumption of GNs, or UAV, or the weighted sum of energy consumption of IoT devices and UAV were investigated in [15–17] with the consideration of time-sensitivity for task offloading, respectively. Meanwhile, to minimize the maximum task latency, authors in [18] designed a novel penalty dual decomposition-based algorithm by jointly optimizing UAV trajectory and task offloading. Due to the limited battery capacity and computation resources of IoT devices, the tradeoff between the energy consumption and time sensitively has also attracted significant attention [19–22]. More specifically, authors in [19] utilized the modified genetic algorithm NSGA-II to solve a multi-objective problem of GNs' average energy consumption and latency time, while the authors in [21] aimed at minimizing the weighted sum of the service delay of all IoT devices and UAV energy consumption. Besides, an alternative optimization algorithm based on successive convex approximation (SCA) was derived to minimize UAV energy and completion time by jointly optimizing computing offloading, resource allocation and trajectory in [22].

Ordinarily, an application consists of several divisible and logically independent tasks. For example, an AR application consists of five critical tasks: video source, tracker, mapper, object recognizer, and renderer. Among these tasks, the tracker, mapper, and objective recognizer are computation-intensive components which can be offloaded to MEC servers to compute or store for follow-up processing [23]. To our knowledge, larger amounts of data and tasks are generating and waiting for proceeding with the explosive growth of smart terminals in the era of big data. To meet the challenge, data caching can be regarded as an effective way to deal with these tasks, and efficiently tackle the following two dilemmas: (1) strong channel gain but insufficient computation capability at UAV, the GN could have offloaded more data to ease its burden of computing; (2) weak channel gain with sufficient computation capability at UAV, the UAV could have computed more for GNs to improve the performance of networks. With the help of data caching, we can foster efficiency in the use of channel and computation capability at UAV concurrently. To be specific, although UAV is engaged in calculation for other GNs at the moment, GN can transmit more data in virtue of better channel gain.

Furthermore, these abundant data or tasks can be buffered first, and wait for processing till UAV has sufficient computation resources. Meanwhile, the computation capability will not be wasted at the situation that GN has low-rate transmission due to poor channel gain, UAV can process the cached data instead.

Nevertheless, these pioneering works above just investigated the single task of IoT devices, and focused on either data collection [11, 12] or computing offloading [14–19, 21, 22], but ignored the effectiveness of data caching. In the meantime, time-sensitivity has been one of most important metric for network performance, and triggered a great deal of research interests in applications related with UAV. Therefore, in this paper, collaborative multi-task computation and caching offloading for UAV-enabled MEC networks is investigated to minimize the total energy consumption of GNs,[1] where the trajectory along with communication and computing resource allocation at UAV is optimized, and task offloading decision at GNs is determined, to satisfy the QoE requirement of time-sensitive tasks of GNs. The main contributions of this paper are summarized as follows:

- Collaborative multi-task computation and caching offloading are considered to take full advantage of communication and computation resources.
- To our knowledge, it is the first attempt to investigate the energy consumption associated with data caching operations, and consider the collaborative offloading.
- Since the applications generated by GNs can be divided into several tasks, the QoE requirement of latency is considered for both the single task and the whole application of GNs.

2 System model and problem formulation

A UAV-assisted MEC system with multiple IoT devices denoted by $\mathcal{K} = \{1, 2, \ldots, K\}$ is considered based on frequency division multiple access (FDMA), as shown in Fig. 1, where the UAV is equipped with an embedded MEC server providing computing resource and data storage service to IoT devices during its navigation. The origin and destination points are denoted by $\mathbf{Q_0}$ and $\mathbf{Q_F}$[2] respectively, and the period of the whole navigation between $\mathbf{Q_0}$ and $\mathbf{Q_F}$ is given by T, which is further divided into N time slots belongs to the set $\mathcal{N}$, and the length of each is τ, $\tau = T/N$. In this system, the three-dimensional Cartesian coordinate system is employed, where the UAV is launched from the $\mathbf{Q_0}$ at a fixed altitude H and flies to the final point $\mathbf{Q_F}$ in the end according to the coordinate series $\boldsymbol{q}[n] = (q_x[n], q_y[n]) \in \mathbb{R}^{2\times 1}, \forall n \in \mathcal{N}$, meanwhile, each IoT device maintains stationary coordinate $\boldsymbol{w}_k = (x_k, y_k), \forall k$. Denote $d_k[n] = \sqrt{(q_x[n] - x_k)^2 + (q_y[n] - y_k)^2 + H^2}$ as the distance between the UAV and the kth IoT device, which varies with respect to (w.r.t.) each slot. Since τ is small, the average velocity between $\boldsymbol{q}[n+1]$ and $\boldsymbol{q}[n]$ can be approximated by the instant speed $\|\boldsymbol{v}[n]\|$, given by

[1] For simplicity, we use GNs instead of IoT devices in the following part.

[2] The origin and destination points can be the same or different point(s).

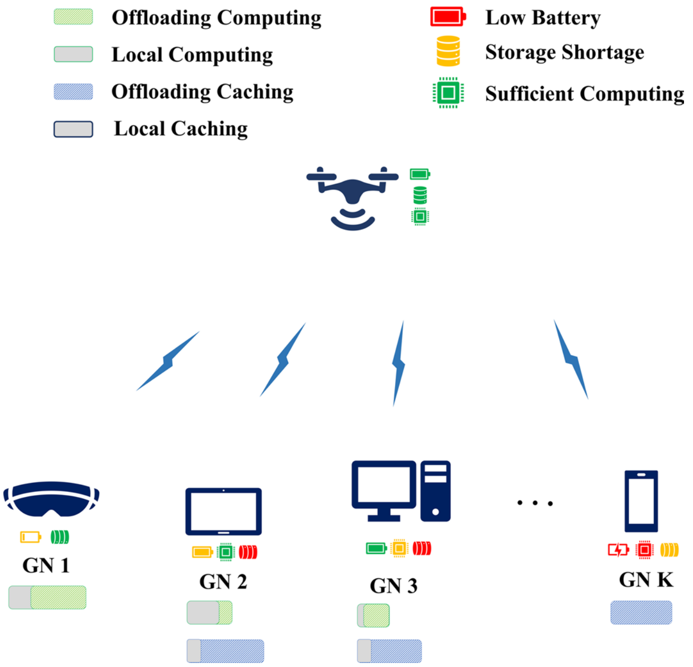

Fig. 1 A UAV-enabled MEC network providing computing and caching offloading services through FDMA mode

$$v[n] \triangleq \|\boldsymbol{v}[n]\| = \frac{\|\boldsymbol{q}[n+1]-\boldsymbol{q}[n]\|}{\tau}, \tag{1}$$

where $\|\cdot\|$ denotes the Euclidean distance, and $v[n]$ cannot exceed its maximum speed $v_{\max}$, $\|\boldsymbol{q}[n+1]-\boldsymbol{q}[n]\| \le \tau v_{\max}$.

Assume that each IoT device has an energy-consuming and delay-sensitive application to be executed within T_k, which defines the kth GN whole mission delay and must not exceed the mission period T, given as $T_k \le T, \forall k$. Herein, partition-oriented applications are investigated, such as virus scan application and figure compression application [24], and the application can be divided into N logically independent tasks w.r.t. the time slots. Furthermore, each logically independent task can be arbitrarily divided into several partitions to support parallelism, and thus handled locally and at the UAV concurrently since the amount of data to be processed is known in advance for this kind of applications [23]. To be specific, the kth IoT device generates a task described by a tuple of five parameters $\{s_k[n], a_k[n], \theta_k, l_k[n], \tau_k[n]\}$ at the nth time slot, where $s_k[n]$ indicates the amount of input data to be processed, $a_k[n]$ is the indicator of task type, i.e., $a_k[n]=1$ for computing, $a_k[n]=0$ for caching, θ_k represents the number of CPU cycles for computing 1-bit of input data, $l_k[n] \in [0,1]$ is the proportion of $s_k[n]$ offloading to UAV, while the rest $(1-l_k[n])s_k[n]$ bits are processed at local, and $\tau_k[n]$ is the maximum tolerable latency shorter than τ. Denote C_k and C as the storage capacity of the kth IoT device and the UAV in bits, respectively. Due to hardware limitations, the total storage resource assigned to IoT devices must satisfy $\sum_{n=1}^{N}\sum_{k=1}^{K}(1-a_k[n])l_k[n]s_k[n] \le C$, and the data cached in the kth IoT device must not exceed its storage limitation, $\sum_{n=1}^{N}(1-a_k[n])(1-l_k[n])s_k[n] \le C_k$. Meanwhile, the local CPU frequency of the kth IoT device is characterized by f_k, and $f_{k,u}[n]$ is the computing resource allocated to the kth IoT device by the UAV at the nth time slot, where both of them are measured by the number of CPU cycles per second. Moreover, a practical constraint that the total computing resources allocated to all the associated IoT devices cannot excess the UAV's computation capability F, is given by $\sum_{k\in\mathcal{K}} a_k[n]f_{k,u}[n] \leq F$.

2.1 Task latency

Suppose that the application generated by the kth GN can be divided into several tasks, which are scheduled into N slots and further partially offloaded to the UAV. With the consideration of time-sensitivity, we assume there are QoE requirements for the single task and the application, respectively. Denote $t_k[n]$ as the latency for the kth GN at nth slot, and the one for the application of kth GN must not exceed T_k, given as, $\sum_{n=1}^{N} t_k[n] \leq T_k, \forall k$, the details of which are given in the following.

2.1.1 Local computing mode

By assuming there is no further operation required for caching the residual part locally, we consider the local computing for executing the residual part $(1 - l_k[n])s_k[n]$ with computation capability f_k at the kth IoT device, the time consumption of which is given by

$$t_k^{\mathrm{loc}}[n] = \frac{a_k[n](1 - l_k[n])s_k[n]\theta_k}{f_k}. \tag{2}$$

2.1.2 Offloading mode

It is assumed that the wireless channel between the UAV and the kth IoT device is dominated by line-of-sight link [25], and the channel between the UAV and IoT devices is modeled by the free space path loss model. Therefore, the channel gain $h_k[n]$ from the kth IoT device to the UAV at the nth time slot is given

$$h_k[n] = \frac{\rho_0}{d_k[n]^2} = \frac{\rho_0}{\|\boldsymbol{q}[n] - \boldsymbol{w}_k\|^2 + H^2}, \tag{3}$$

where ρ_0 represents the channel gain at the reference distance $d_0 = 1$ meter. Since the change of UAV position is negligible when the LoS link path is applied, the channel gain remains constant within each slot. Assume that the total available bandwidth B is shared among IoT devices with a proportion $Bb_k[n]$ for each during a time slot, and the sum of the bandwidth allocated to IoT devices cannot exceed the total bandwidth, $\sum_{k=1}^{K} b_k[n] \leq 1, \forall n$. Thus, the achievable offloading rate at the kth IoT device is given as

$$r_k[n] = Bb_k[n]\log_2\left(1 + \frac{P_k h_k[n]}{Bb_k[n]N_0}\right), \tag{4}$$

where P_k denotes the transmit power of the kth IoT device, and N_0 is the noise power dense at the UAV.

The offloading time $t_k^{\text{off}}[n]$ of the kth IoT device mainly consists of two parts[3]: the uplink transmission time $t_k^{\text{up}}[n]$ from the IoT device to the MEC-integrated BS, and the corresponding execution time at MEC server $t_k^{\text{ex}}[n]$ for the computing and the caching task, respectively. It's assumed that there is no further processing for cached data at UAV and the operation will always succeed. The caching time is negligible and the result of caching operation need not be returned. Thus for the offloading caching task, we only take the task transmission time into account. Therefore, the offloading time for both the computing and the caching task can be written as

$$\begin{aligned} t_k^{\text{off}}[n] &= t_k^{\text{up}}[n] + t_k^{\text{ex}}[n] \\ &= \frac{l_k[n]s_k[n]}{r_k[n]} + \frac{a_k[n]l_k[n]s_k[n]\theta_k}{f_{k,u}[n]}. \end{aligned} \tag{5}$$

Due to the parallel computing at IoT devices and the UAV, the total latency for the kth IoT device depends on the larger one between $t_k^{\text{loc}}[n]$ and $t_k^{\text{off}}[n]$, which can be recast as $t_k[n] \triangleq \max\{t_k^{\text{loc}}[n], t_k^{\text{off}}[n]\}, \forall k, n.$

2.2 Energy consumption

Since GN tasks can be computed or cached locally or in UAV, both the computing and caching operation are charged for energy consumption. Similar to [26], the in-memory caching energy consumption model is adopted for both UAV and GN, in which the energy consumption is proportional to the cached task bits. Additionally, the signal transmission consumption of GNs and UAV propulsion energy are also included.

2.2.1 Energy consumption at the IoT device

The energy consumed by the kth IoT device includes the cost of uploading and the part of local operation, which varies according to corresponding task type. For a local computing task, the power consumption of the processor at the kth IoT device is modeled as $\eta_k f_k^3$ (joule per second), given as,

$$E_k^{\text{exe}}[n] = a_k[n]\eta_k(1 - l_k[n])s_k[n]\theta_k f_k^2, \tag{6}$$

where η_k represents the computation energy efficiency coefficient related to the processor's chip. On the other hand, based on the energy-proportional model the local caching energy consumed for processing $(1 - l_k[n])s_k[n]$ bits data at kth GN can be written as

$$E_k^{\text{ca}}[n] = (1 - a_k[n])(1 - l_k[n])\omega_k^{ca} s_k[n], \tag{7}$$

where ω_k^{ca} is the coefficient of caching power efficiency related to the GN caching hardware drives [27–29]. With the help of (5), the transmission energy consumption at the kth IoT device is given as

$$E_k^{\text{tran}}[n] = \frac{P_k l_k[n]s_k[n]}{r_k[n]}. \tag{8}$$

[3] In practice, the amount of output data from MEC server to the kth GN is usually much less than that of the input data, and thus the time consumed and the transmission energy for delivering the computed results are negligible [10].

In the end, the total energy consumption of kth GN during flight navigation can be written as follows,

$$E_k = \sum_{n=1}^{N} \left(E_k^{\text{ca}}[n] + E_k^{\text{exe}}[n] + E_k^{\text{tran}}[n] \right), \forall k. \tag{9}$$

2.2.2 UAV energy consumption

During the entire flight, the energy consumption for propulsion and task processing are involved at UAV. The part for offloaded computing task at UAV is similarly given as

$$E_{\text{exe}} = \sum_{n=1}^{N} \sum_{k=1}^{K} a_k[n] \eta l_k[n] s_k[n] \theta f_{k,u}^2[n], \tag{10}$$

where η denotes the computation energy efficiency coefficient related to the processor's chip of UAV. However, if the offloaded task is claimed as a caching task, the caching energy can be similarly written as

$$E_{\text{ca}} = \omega_{\text{ca}} \sum_{n=1}^{N} \sum_{k=1}^{K} (1 - a_k[n]) l_k[n] s_k[n], \tag{11}$$

where ω_{ca} is the caching energy coefficient related to UAV storage hardware. Additionally, similar to [10, 12], the rotary-wing energy model is adopted for UAV propulsion energy consumption.

$$E_{\text{fly}} = \sum_{n=1}^{N} \tau \left[P_0 \left(1 + \frac{3v^2[n]}{Q_{\text{tip}}^2} \right) + \frac{1}{2} d_0 s \rho A v^3[n] + P_i \left(\sqrt{1 + \frac{v^4[n]}{4v_0^4}} - \frac{v^2[n]}{2v_0^2} \right)^{\frac{1}{2}} \right], \tag{12}$$

where P_0 and P_i denote the constant *blade profile power* and *induced power* in hovering status. For other parameters, Q_{tip} is the tip speed of the rotor blade, v_0 is the mean rotor induced velocity in hover, d_0 and s are the fuselage drag ratio and rotor solidity, and ρ and A define the air density and rotor disc area.

2.3 Problem formulation

In this paper, we jointly optimize trajectory $\boldsymbol{q}$, bandwidth $\boldsymbol{b}$, computation capability $\boldsymbol{f}$, task type $\boldsymbol{a}$ and offloading ratio $\boldsymbol{l}$, to minimize the total energy consumption of IoT devices in the following problem,

$$\min_{\substack{\boldsymbol{q},\boldsymbol{b},\boldsymbol{f} \\ \boldsymbol{a},\boldsymbol{l}}} \sum_{k=1}^{K} E_k \tag{13a}$$

$$\text{s.t.} \sum_{k=1}^{K} b_k[n] \leq 1, \ \forall n,$$

$$\mathbf{P}: \sum_{k=1}^{K} a_k[n] f_{k,u}[n] \le F, \forall n, \tag{13b}$$

$$\sum_{n=1}^{N}\sum_{k=1}^{K}(1-a_k[n]) l_k[n] s_k[n] \le C, \tag{13c}$$

$$\sum_{n=1}^{N}(1-a_k[n])(1-l_k[n]) s_k[n] \le C_k, \forall k, \tag{13d}$$

$$E_{\text{fly}} + E_{\text{exe}} + E_{\text{ca}} \le E, \tag{13e}$$

$$t_k[n] \le \tau_k[n], \ \forall k, n, \tag{13f}$$

$$\sum_{n=1}^{N} t_k[n] \le T_k, \ \forall k, \tag{13g}$$

$$\boldsymbol{q}[0] = \mathbf{Q_0}, \boldsymbol{q}[N] = \mathbf{Q_F}, \tag{13h}$$

$$\|\boldsymbol{q}[n+1] - \boldsymbol{q}[n]\| \le \tau v_{\max}, \ \forall n, \tag{13i}$$

$$0 \le b_k[n] \le 1, \ \forall k, n, \tag{13j}$$

$$0 \le l_k[n] \le 1, \ \forall k, n, \tag{13k}$$

$$a_k[n] \in \{0,1\}, \ \forall k, n, \tag{13l}$$

where $\boldsymbol{q} \triangleq \{\boldsymbol{q}[n]\}$, $\boldsymbol{b} \triangleq \{b[n]\}$, $\boldsymbol{a} \triangleq \{a_k[n]\}$, $\boldsymbol{l} \triangleq \{l_k[n]\}$ and $\boldsymbol{f} \triangleq \{f_{k,u}[n]\}, \forall k, n$. Constraints (13a)–(13d) are the resource limitations introduced above. Constraint (13e) indicates that the total energy consumption must not exceed UAV battery capacity. The QoE requirements for the single task and the application are denoted by (13f) and (13g), respectively, ensuring the instantaneous and long-time performance for GNs.

3 Proposed solution and algorithm

Obviously, **P** is a challenging mixed integer nonlinear programming (MINLP) problem due to the coupled variants in both the constraints and the objective function. Instead of dealing with the original problem directly, we divide **P** into three manageable subproblems to decouple these variants: (1) Trajectory optimization for $\boldsymbol{q}$; (2) Resource allocation at UAV including bandwidth $\boldsymbol{b}$ and computation capability $\boldsymbol{f}$; (3) Offloading decisions including task type $\boldsymbol{a}$ and ratio $\boldsymbol{l}$. The iterative approach based on the block coordinate descent (BCD) method [30, 31] is thus employed to alternately solve the three subproblems.

3.1 Trajectory optimization

Given $\{\boldsymbol{b},\boldsymbol{f},\boldsymbol{a},\boldsymbol{l}\}$, the trajectory of UAV can be derived according to the following problem,

$$\begin{aligned} \mathbf{P_T}: \min_{\boldsymbol{q}} & \sum_{n=1}^{N}\sum_{k=1}^{K}\frac{P_k l_k[n] s_k[n]}{r_k[n]} \\ & \text{s.t. } (13e)-(13i), \end{aligned} \tag{14}$$

which is still intractable due to the non-convexity of constraints (13e)–(13g), as well as the objective function.

Motivated by [12, 17], the SCA technique is utilized to cope with the non-convexity with introduced auxiliary variables $\boldsymbol{o} \triangleq \{o[n] \geq 0\}$, where $o[n] = \left(\sqrt{1+\frac{v^4[n]}{4v_0^4}} - \frac{v^2[n]}{2v_0^2}\right)^{\frac{1}{2}}$. Thus, constraint (13e) can be recast as

$$\sum_{n=1}^{N}\tau\left[P_0\left(1+\frac{3v^2[n]}{q^2}\right) + P_i o[n] + \frac{1}{2}d_0 s\rho A v^3[n]\right] + E_{\text{exe}} + E_{\text{ca}} \leq E, \tag{15a}$$

$$\frac{1}{o^2[n]} \leq o^2[n] + \frac{v^2[n]}{v_0^2}, \tag{15b}$$

where constraint (15b) is still non-convex. To cope with the non-convexity, we derive the following inequalities by employing the first-order Taylor expansion with given points $(o_i[n], v_i[n])$ at the ith iteration,

$$\frac{1}{o^2[n]} \leq \left(2o_i[n]o[n] - o_i[n]^2\right) + \left(2v_i[n]v[n] - v_i^2[n]\right) \tag{16}$$

where the right-hand side of (16) is a linear function w.r.t. $v[n]$ and $o[n]$. Similarly, auxiliary variables $\tilde{r}_k^q[n] \geq \frac{1}{r_k[n]} > 0, \forall n$ are introduced to deal with non-convex parts related to $r_k[n]$ in the objective function and constraints (13f) and (13g). To be specific, for any given local point $\tilde{r}_{k,i}^q[n]$, a lower bound function named as $L(\tilde{r}_{k,i}^q[n])$ is defined to transform $r_k[n]$ as follows

$$L(\tilde{r}_k^q[n]) \triangleq \phi_0\left[\frac{1}{\omega_k^i - 1} - \frac{\omega_k - \omega_k^i}{\left(\omega_k^i - 1\right)^2}\right], \tag{17}$$

where $\omega_k^i \triangleq 2^{(Bb_k[n]\tilde{r}_{k,i}^q[n])^{-1}}$, $\omega_k \triangleq 2^{\left(Bb_k[n]\tilde{r}_k^q[n]\right)^{-1}}$ and $\phi_0 = \frac{p_k[n]\rho_0}{Bb_k[n]N_0}$. Hence, constraints (13f) and (13g) can be equivalently rewritten as

$$\frac{a_k[n]l_k[n]s_k[n]\theta}{f_{k,u}[n]} + l_k[n]s_k[n]\tilde{r}_k^q[n] \leq \tau_k[n], \tag{18a}$$

$$\|\boldsymbol{q}[n] - \boldsymbol{w}_k\|^2 + H^2 \leq L(\tilde{r}_k^q[n]), \tag{18b}$$

$$\sum_{n=1}^{N} \max\left\{ t_k^{loc}[n], \frac{a_k[n]l_k[n]s_k[n]\theta}{f_{k,u}[n]} + l_k[n]s_k[n]\tilde{r}_k^q[n] \right\} \leq T_k. \tag{18c}$$

The lower bound function $L(\tilde{r}_k^q[n])$ is concave w.r.t. $\tilde{r}_k^q[n]$ satisfying $\frac{\phi_0}{2^{(Bb_k[n]\tilde{r}_k^q[n])^{-1}}-1} \geq L(\tilde{r}_k^q[n])$ with the given local point at the ith iteration, the proof of which can be found in [17]. Therefore, $\mathbf{P_T}$ can be transformed into a convex optimization problem as

$$\begin{aligned} &\min_{\{\boldsymbol{q},\boldsymbol{v},\boldsymbol{o},\tilde{\boldsymbol{r}}_{\boldsymbol{q}}\}} \sum_{n=1}^{N}\sum_{k=1}^{K} p_k l_k[n]s_k[n]\tilde{r}_k^q[n] \\ \mathbf{P_{T1}}:\ &\text{s.t. } (13h)(13i)(15a)(16)(18a)(18b)(18c), \end{aligned} \tag{19}$$

where $\tilde{\boldsymbol{r}}_{\boldsymbol{q}} \triangleq \{\tilde{r}_k^q[n]\}$, and the details of solving $\mathbf{P_{T1}}$ is presented in Algorithm 1.

Algorithm 1 Algorithm for Trajectory Optimization

```
Input: Give initial points $\{\boldsymbol{q}_i, \boldsymbol{b}_i, \boldsymbol{f}_i, \boldsymbol{a}_i, \boldsymbol{l}_i\}$, and set $i = 0$.
Output: $\boldsymbol{q}^\star$.
initialize: $v_i[n] = \|\boldsymbol{q}_i[n+1] - \boldsymbol{q}_i[n]\|/\tau$, $o_i[n] = \left(\sqrt{(1+\frac{\boldsymbol{v}_i[n]^4}{4v_0^4})} - \frac{\boldsymbol{v}_i[n]^2}{2v_0^2}\right)^{\frac{1}{2}}$ and $\tilde{\boldsymbol{r}}^q[n] = \{\tilde{r}_{k,i}^q[n], \forall k\}, \forall n$.
repeat
    with given local points $\{\boldsymbol{q}_i[n], o_i[n], \boldsymbol{v}_i[n], \tilde{r}_{k,i}^q[n]\}$, obtain optimal solutions $\{\boldsymbol{q}^\star[n], o^\star[n], \boldsymbol{v}^\star[n], \tilde{r}_q^\star[n]\}$ by solving $\mathbf{P_T1}$.
    update local points $\{\boldsymbol{q}_{i+1}[n] = \boldsymbol{q}^\star[n], o_{i+1}[n] = o^\star[n], v_{i+1}[n] = v^\star[n], \tilde{r}_{k,i+1}^q[n] = \tilde{r}_q^\star[n]\}$.
until converge to the tolerant accuracy.
```

3.2 Resource allocation at UAV

With the newly obtained $\boldsymbol{q}^\star$ and the given $\{\boldsymbol{a}, \boldsymbol{l}\}$, the resource allocation strategy at UAV can be designed via the following problem,

$$\begin{aligned} &\min_{\{\boldsymbol{b},\boldsymbol{f}\}} \sum_{n=1}^{N}\sum_{k=1}^{K} \frac{P_k l_k[n]s_k[n]}{r_k[n]} \\ \mathbf{P_R}:\ &\text{s.t } \frac{1}{r_k[n]} + \frac{a_k[n]\theta}{f_{k,u}[n]} \leq \frac{\tau_k[n]}{l_k[n]s_k[n]}, \forall k, n, \\ &(13a)(13b)(13e)(13g)(13j) \end{aligned} \tag{20a}$$

which is non-convex due to the nonlinear part in the objective function and constraints (13g) and (20a), i.e., $r_k[n]$. Similarly, the auxiliary variables $\tilde{\boldsymbol{r}}_{\boldsymbol{b}} \triangleq \{0 < \tilde{r}_k^b[n] \leq r_k[n], \forall k, n\}$ are introduced to deal with $r_k[n]$, which further transforms $\mathbf{P_R}$ into the following equivalent problem,

$$\begin{aligned} &\min_{\{\boldsymbol{b},\boldsymbol{f},\tilde{\boldsymbol{r}}_{\boldsymbol{b}}\}} \sum_{n=1}^{N}\sum_{k=1}^{K} \frac{P_k l_k[n]s_k[n]}{\tilde{r}_k^b[n]} \\ &\text{s.t. } (13a)(13b)(13j) \\ \mathbf{P_{R1}}:\ &0 < \tilde{r}_k^b[n], \forall k, n, \end{aligned} \tag{21a}$$

$$\tilde{r}_k^b[n] \le Bb_k[n] \log_2 (1 + \frac{\phi_1}{Bb_k[n]}), \forall k, n, \tag{21b}$$

$$\sum_{n=1}^{N}\sum_{k=1}^{K} a_k[n]\eta l_k[n] s_k[n]\theta f_{k,u}^2[n] \le E', \tag{21c}$$

$$\frac{a_k[n]\theta}{f_{k,u}[n]} + \frac{1}{\tilde{r}_k^b[n]} \le \frac{\tau_k[n]}{l_k[n]s_k[n]}, \forall k, n, \tag{21d}$$

$$\sum_{n=1}^{N} \max\left\{ t_k^{loc}[n], \frac{a_k[n]l_k[n]s_k[n]\theta}{f_{k,u}[n]} + \frac{l_k[n]s_k[n]}{\tilde{r}_k^b[n]} \right\} \le T_k, \tag{21e}$$

where E' is a constant value related to energy consumption. Define ϕ_1 as $\phi_1 \triangleq \frac{P_k\rho_0}{(\|\boldsymbol{q}[n]-\boldsymbol{w}_k\|^2+H^2)N_0}$ and the second-order derivative of the right side in constraint (21b) is given as $\nabla^2 r_k[n](b_k[n]) = -\frac{\phi_1^2}{Bb_k[n](Bb_k[n]+\phi_1)^2 \ln 2}$, which proves the concavity of constraint (21b). It is also observed that (21e) is convex because the second item in max method is convex and the first item is a constant value. Therefore, $\mathbf{P_{R1}}$ is a convex problem w.r.t. $\{\boldsymbol{b}, \boldsymbol{f}, \tilde{\boldsymbol{r}}_{\boldsymbol{b}}\}$, which can be efficiently solved by solvers, e.g., CVX [32].

3.3 Offloading decisions at GNs

In this subsection, the task type and offloading ratio are optimized to further minimize the total energy consumption of MDs with the achieved $\{\boldsymbol{q}^\star, \boldsymbol{b}^\star, \boldsymbol{f}^\star\}$ as follows

$$\begin{aligned} \mathbf{P_O} : & \min_{\{\boldsymbol{a},\boldsymbol{l}\}} \sum_{n=1}^{N}\sum_{k=1}^{K} E_k[n] \\ & \text{s.t. } (13b) - (13g), (13k) - (13l), \end{aligned} \tag{22}$$

which is also difficult to deal with since $\boldsymbol{a}$ consists of binary variables and further couples with $\boldsymbol{l}$. It's observed that with fixed $\boldsymbol{a}$, $\mathbf{P_o}$ is the linear programming problem w.r.t $\boldsymbol{l}$ and given with fixed $\boldsymbol{l}$ the subproblem becomes an integer programming problem **w.r.t** $\boldsymbol{a}$. Thus, an inner BCD algorithm is derived to alternatively solve the offloading problem.

Table 1 Simulation parameters

System parameters	Values
Maximal CPU frequency of UAV	5 GHz
CPU frequency of GN	50 MHz
Total bandwidth B	10 MHz
Max storage capacity C at UAV	200 M
GN remaining storage capacity C_k	0.5–5 MBits
Transmit power of GN	1 W
Task size at GN	100–500 Kbits
Computation intensity θ, θ_k	900–1000 cycles/bit
Background noise N_0	− 130 dBm
Channel gain at $d_0 = 1$m	50 dB
UAV related parameters	**Values**
UAV blade power P_0	158.76 W
Induced power P_i	88.63 W
Blade speed Q_{tip}	120 m/s
Air density ρ	1.225 kg/m^3
Rotor disc area A	1 m^2
Hover induced velocity V_0	4.03
Fuselage drag ratio d_0	0.3
Fuselage rotor solidity s	0.05
UAV maximum velocity $V_{\max}$	15 m/s
UAV flight height H	100 m

Algorithm 2 Proposed Algorithm

```
Input: Give initial points $\{\boldsymbol{q}_i, \boldsymbol{b}_i, \boldsymbol{f}_i, \boldsymbol{a}_i, \boldsymbol{l}_i\}$, and set $i = 0$.
Output: $\{\boldsymbol{q}^\star, \boldsymbol{b}^\star, \boldsymbol{f}^\star, \boldsymbol{a}^\star, \boldsymbol{l}^\star\}$.
repeat
    given $\{\boldsymbol{q}_i, \boldsymbol{b}_i, \boldsymbol{f}_i, \boldsymbol{a}_i, \boldsymbol{l}_i\}$, calculate the trajectory of UAV $\boldsymbol{q}_{i+1}$ based on Algorithm 1.
    given offloading decisions $\{\boldsymbol{a}_i,\ \boldsymbol{l}_i\}$ and $\boldsymbol{q}_{i+1}$, obtain the resource allocation strategy $\{\boldsymbol{b}_{i+1},\ \boldsymbol{f}_{i+1}\}$ at UAV by solving $\mathbf{P_R1}$.
    with the achieved $\{\boldsymbol{q}_{i+1}, \boldsymbol{b}_{i+1}, \boldsymbol{f}_{i+1}\}$, derive the offloading decisions $\{\boldsymbol{a}_{i+1},\ \boldsymbol{l}_{i+1}\}$ by solving $\mathbf{P_O}$.
    update variables: $\boldsymbol{q}_i = \boldsymbol{q}_{i+1}$, $\boldsymbol{b}_i = \boldsymbol{b}_{i+1}$, $\boldsymbol{f}_i = \boldsymbol{f}_{i+1}$, $\boldsymbol{a}_i = \boldsymbol{a}_{i+1}$, $\boldsymbol{l}_i = \boldsymbol{l}_{i+1}$.
until converge to the tolerant accuracy.
```

4 Simulation results and discussions

In this section, several experiments are conducted for verifying the performance of proposed algorithm. Assume that GNs are distributed in a square area of size 100m × 100m and generate multiple tasks during the navigation of UAV, the sizes of which range from 100 to 500 Kbits within each time slot. The UAV starts flight at origin point $\mathbf{Q_0} = (\mathbf{0}, \mathbf{0})$ and follows the optimized trajectory to the final point $\mathbf{Q_F}$, which can be the origin or a different point. The details of the experimental parameters are summarized in Table 1, unless otherwise mentioned. Numerical results are presented for the UAV trajectory and the energy consumption of GNs w.r.t different offloading schemes: (1) **Computing offloading mode**: all of GNs' tasks are for computing; (2)

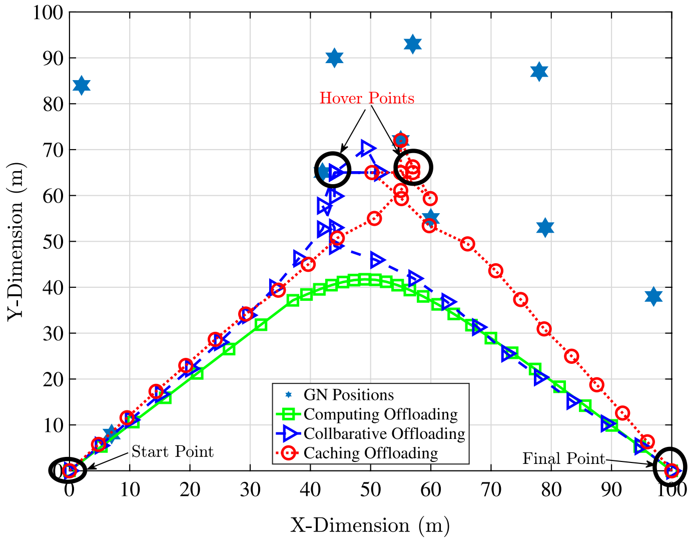

Fig. 2 The UAV trajectories w.r.t. different offloading schemes when $\tau = 0.5$ s

Collaborative offloading mode: the tasks of GNs are for either computing or caching; (3) **Caching offloading mode**: all the tasks are only for caching at UAV or at GNs.

In Fig. 2, the trajectory of UAV w.r.t. different offloading modes is investigated, where 10 GNs are stochastically distributed and generate tasks with size ranging from 100 to 200 Kbits. The total slot numbers is $N = 20$, the length of which is 0.5 s. Compared with computing and collaborative offloading mode, caching offloading mode authorizes UAV to get closer to GNs, since it does not need to process the offloaded data, gives UAV enough time to approach GNs and further gets better communication quality in return, which is beneficial to energy saving as illustrated in Fig. 5. Since collaborative offloading mode is the combination of the other two modes, it is observed that UAV can get relatively closer to GNs at that mode, hover at some points to render services, and further save energy just like that of caching offloading mode.

In Fig. 3, the trajectory of UAV with different final points are investigated, and the optimized UAV trajectories can vary according to its final points under the same environment settings. Notice that the first half trajectories for four different final points appear to overlap, and the flight path is a straight line when the UAV ends at the origin and point (100, 100), while the UAV made a slow arcs to GNs at the second half trajectory for the other two final points, i.e., (0, 100) and (100, 0). It's mainly because UAV wants to fly closer to GNs for stronger channel gain, thus provides them with better services.

In Fig. 4, with the same environment settings, the trajectory of UAV w.r.t. different QoE requirement of time-sensitive tasks are presented. Along with the decreasing QoE requirement of latency, the trajectory will gradually fly towards the GNs, since it has more time to approach GNs, and further offer better services as explained above. Notice that, when the QoE requirement of latency exceeds a certain limitation, i.e., $\tau = 0.5$ s, GNs can easily accomplish its task with the constraint of QoE requirement

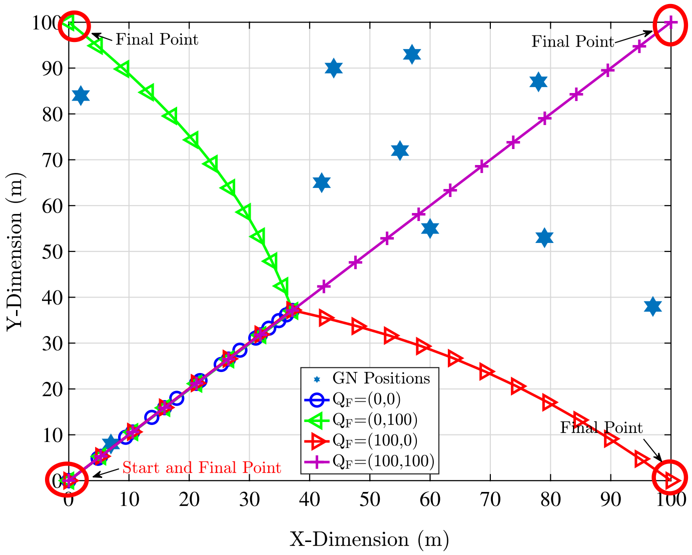

Fig. 3 The UAV trajectories w.r.t. different final points with $\tau = 0.5$ s

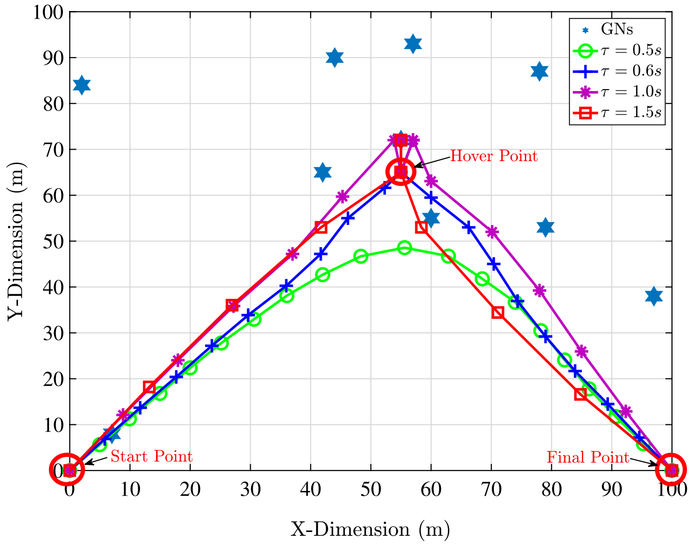

Fig. 4 The UAV trajectories w.r.t. different QoE requirement of time-sensitive tasks at collaborative offloading mode

for latency, and thus the UAV will not fly closer to UE due to its purpose of energy saving. Besides, the trajectory (the red and pink lines) indicates that the UAV will hover at some points during the navigation to save its energy, rather than moving forward.

In Fig. 5, we investigate the performance of the total energy consumption of GNs for different schemes. Due to the different energy consumption and operation for computing and caching task, GNs at **computing offloading mode** consume more energy compared with collaborative offloading and caching offloading schemes, since all the tasks are designed for computing that consumes much energy than caching. Meanwhile, the

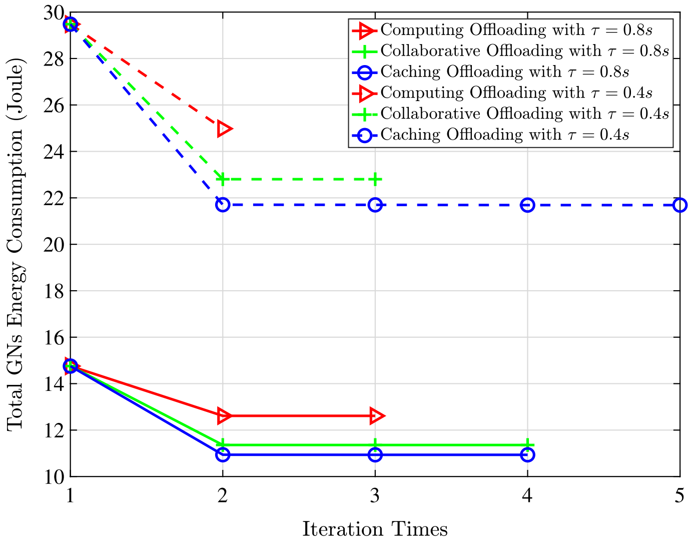

Fig. 5 The total energy consumption of GNs w.r.t. different offloading schemes when $K = 6$

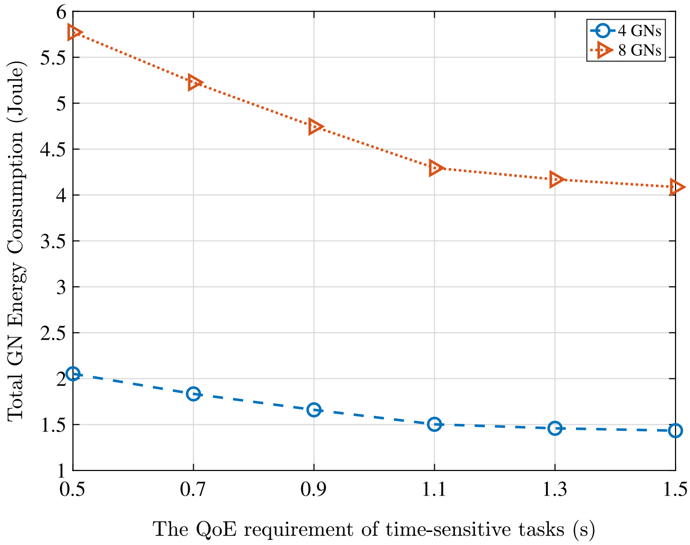

Fig. 6 The total energy consumption of different numbers of GNs w.r.t. different QoE requirement of time-sensitive tasks

caching offloading mode is the most energy-saving since all the tasks are for caching, and thus consume less energy compared with the others. Furthermore, **collaborative offloading scheme** not only meets the demands for different type of tasks, but also gets relatively satisfactory result on energy saving, since it can switch parts of tasks into caching.

In Fig. 6, we study the influence of the QoE requirements of time-sensitive tasks at IoT devices on the total energy consumption for different numbers of GNs. With the growing number of GNs, the total energy consumption of GNs is increasing for all the schemes. On the observation of Fig. 6, the total energy consumption is reducing along with the decreasing QoE requirement of time-sensitive tasks. This is because UAV has sufficient time to get closer to GNs with the decreasing requirement of latency, providing better communication access (i.e., better channel gain), and thus help to reduce the total energy consumption of GNs.

5 Conclusions

With the consideration of QoE requirement of time-sensitive tasks, a UAV-assisted multi-task MEC networks is investigated, where UAV provides collaborative offloading services to GNs. We jointly optimize the trajectory, resource allocation of UAV, and offloading decisions of GNs to minimize the total energy consumption of GNs according to BCD method. Simulation results demonstrate that collaborative computing and caching offloading can effectively reduce the total energy consumption of GNs, while satisfying the QoE demands for different type of tasks.

Abbreviations

AR: Augmented reality; BS: Base station; GN: Ground nodes; IoT: Internet of things; LoS: Line of sight; MEC: Mobile edge computing; SN: Sensor nodes; QoE: Quality of experience; QoS: Quality of service; UAV: Unmanned aerial vehicle; VR: Virtual reality.

Acknowledgements

We would like to thank the generous support of Engineering Research Center of Cyberspace, and the Key Laboratory in Software Engineering of Yunnan Province.

Authors' contributions

W. Li and M. Zhao contributed to the development of the system model, design of the study, theory, result analysis, and article writing. W. Li conceived, designed, and performed the experiments. Y. Wu and M. Zhao optimized the proposed models and undertook revision works of the paper. All authors read and approved the final manuscript.

Funding

This work is supported in part by the National Natural Science Foundation of China under Grant 61801418 and 61902341, in part by Yunnan Applied Basic Research Projects 2019FD-129, 202001BB050034 and 202001BB050035.

Availability of data and materials

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Competing interests

The authors declare that they have no competing interests.

Received: 30 August 2020 Accepted: 6 November 2020
Published online: 06 January 2021

References

1. C. Xie, B. Yu, Z. Zeng, Y. Yang, Q. Liu, Multi-layer internet of things middleware based on knowledge graph. IEEE Internet Things J. (2020). https://doi.org/10.1109/JIOT.2020.3019707
2. 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**(4), 2322–2358 (2017)
3. C. She, C. Sun, Z. Gu, Y. Li, C. Yang, H.V. Poor, B. Vucetic, A tutorial of ultra-reliable and low-latency communications in 6G: integrating theoretical knowledge into deep learning. arXiv:2009.06010. Accessed 13 Oct 2020
4. N. Abbas, Y. Zhang, A. Taherkordi, T. Skeie, Mobile edge computing: a survey. IEEE Internet Things J. **5**(1), 450–465 (2018)
5. P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, T. Taleb, Survey on multi-access edge computing for internet of things realization. IEEE Commun. Surv. Tutor. **20**(4), 2961–2991 (2018)
6. J.J. Yu, M. Zhao, W.T. Li, D. Liu, W. Feng, Joint Offloading and resource allocation for time-sensitive multi-access edge computing network. In: *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6 (2020)
7. M. Zhao, J. Yu, W. Li, D. Liu, S. Yao, W. Feng, C. She, T.Q.S. Quek, Energy-aware offloading in time-sensitive networks with mobile edge computing. arXiv:abs/2003.12719 (2020)
8. J.-J. Yu, H. Wang, M. Zhao, W. Li, H. Bao, L. Yin, M. Wu, Energy minimization for mobile edge computing networks with time-sensitive constraints. arXiv:abs/2008.09504 (2020)
9. M. Mozaffari, W. Saad, M. Bennis, Y. Nam, M. Debbah, A tutorial on UAVs for wireless networks: applications, challenges, and open problems. IEEE Commun. Surv. Tutor. **21**(3), 2334–2360 (2019)

10. Y. Zeng, J. Xu, R. Zhang, Energy minimization for wireless communication with rotary-wing UAV. IEEE Trans. Wirel. Commun. **18**(4), 2329–2345 (2019)
11. C. Zhan, Y. Zeng, R. Zhang, Energy-efficient data collection in UAV enabled wireless sensor network. IEEE Wirel. Commun. Lett. **7**(3), 328–331 (2018)
12. C. Zhan, Y. Zeng, Aerial-ground cost tradeoff for multi-UAV-enabled data collection in wireless sensor networks. IEEE Trans. Commun. **68**(3), 1937–1950 (2020)
13. C. She, C. Yang, T.Q.S. Quek, Radio resource management for ultra-reliable and low-latency communications. IEEE Commun. Mag. **55**(6), 72–78 (2017)
14. T. Zhang, Y. Xu, J. Loo, D. Yang, L. Xiao, Joint computation and communication design for UAV-assisted mobile edge computing in IoT. IEEE Trans. Ind. Inform. **16**(8), 5505–5516 (2020)
15. H. Guo, J. Liu, UAV-enhanced intelligent offloading for internet of things at the edge. IEEE Trans. Ind. Inf. **16**(4), 2737–2746 (2020)
16. T. Zhang, Y. Xu, J. Loo, D. Yang, L. Xiao, Joint computation and communication design for UAV-assisted mobile edge computing in IoT. IEEE Trans. Ind. Inf. **16**(8), 5505–5516 (2020)
17. 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**(12), 2357–2361 (2019)
18. 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 Things J. **6**(2), 1879–1892 (2019)
19. L. Cui, C. Xu, S. Yang, J.Z. Huang, J. Li, X. Wang, Z. Ming, N. Lu, Joint optimization of energy consumption and latency in mobile edge computing for internet of things. IEEE Internet Things J. **6**(3), 4791–4803 (2018)
20. Z. Zhao, S. Bu, T. Zhao, Z. Yin, M. Peng, Z. Ding, T.Q.S. Quek, On the design of computation offloading in fog radio access networks. IEEE Trans. Veh. Technol. **68**(7), 7136–7149 (2019)
21. Z. Yu, Y. Gong, S. Gong, Y. Guo, Joint task offloading and resource allocation in UAV-enabled mobile edge computing. IEEE Internet Things J. **7**(4), 3147–3159 (2020)
22. C. Zhan, H. Hu, X. Sui, Z. Liu, D. Niyato, Completion time and energy optimization in UAV-enabled mobile edge computing system. IEEE Internet Things J. **7**(8), 7808–7822 (2020)
23. Y. Dai, D. Xu, S. Maharjan, Y. Zhang, Joint computation offloading and user association in multi-task mobile edge computing. IEEE Trans. Veh. Technol. **67**(12), 12313–12325 (2018)
24. Y. Wang, M. Sheng, X. Wang, L. Wang, J. Li, Mobile-edge computing: partial computation offloading using dynamic voltage scaling. IEEE Trans. Commun. **64**(10), 4268–4282 (2016)
25. Q. Wu, R. Zhang, Common throughput maximization in UAV-enabled OFDMA systems with delay consideration. IEEE Trans. Commun. **66**(12), 6614–6627 (2018)
26. J. Xu, K. Ota, M. Dong, Saving energy on the edge: in-memory caching for multi-tier heterogeneous networks. IEEE Commun. Mag. **56**(5), 102–107 (2018)
27. F. Gabry, V. Bioglio, I. Land, On energy-efficient edge caching in heterogeneous networks. IEEE J. Sel. Areas Commun. **34**(12), 3288–3298 (2016)
28. Jia, Q., Xie, R., Huang, T., Liu, J., Liu, Y.: Energy-efficient cooperative coded caching for heterogeneous small cell networks. In: *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 468–473 (2017)
29. Zhang, X., Ren, Y., Lv, T.: Energy efficiently caching and transmitting scalable videos in HetNets. In: *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6 (2018)
30. M. Zhao, J.Y. Ryu, J. Lee, T.Q. Quek, S. Feng, Exploiting trust degree for multiple-antenna user cooperation. IEEE Trans. Wirel. Commun. **16**(8), 4908–4923 (2017)
31. P. Richtárik, M. Takáč, Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. Math. Program. **144**(1–2), 1–38 (2014)
32. Grant, M., Boyd, S.: CVX: Matlab Software for Disciplined Convex Programming, Version 2.1 (2014)

Publisher's Note

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