Information Sciences 573 (2021) 519–540

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

STMTO: A smart and trust multi-UAV task offloading system

Jialin Guo [a], Guosheng Huang [b], Qiang Li [c], Neal N. Xiong [d,*], Shaobo Zhang [e], Tian Wang [f,g]

[a] School of Computer Science and Engineering, Central South University, Changsha, Hunan 410083, China
[b] School of Information Science and Engineering, Hunan First Normal University, Changsha 410205, China
[c] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China
[d] Department of Mathematics and Computer Science, Northeastern State University, Tahlequah, OK, USA
[e] School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, Hunan, China
[f] Artificial Intelligence and Future Networks, Beijing Normal University & UIC, Zhuhai, Guangdong, China
[g] College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China

ARTICLE INFO

Article history:
Received 31 January 2021
Received in revised form 22 March 2021
Accepted 10 May 2021
Available online 31 May 2021

Keywords:
Multi Unmanned Aerial Vehicles
Task offloading
Trust
Double auction model
Energy effective
Security

ABSTRACT

As one of promising distributed multi-robot system, Unmanned Aerial Vehicles (UAVs) can collaborate to offload complex tasks in edge networks. A Smart and Trust Multi-UAV Task Offloading (STMTO) system is established to offload tasks from Internet of Thing (IoT) devices to edge severs through UAVs with a trust style. In STMTO system, first, a group of UAVs is dispatched to relay tasks from devices to edge servers with rich computing resource. A collaborative task collection scheme is proposed to minimize energy consumption and the task processing delay by dividing working area for each UAV and designing the flight trajectory. Secondly, a many-to-many task double auction model is established for devices and edge servers to maximize the offloading utility, where devices act as buyers, edge servers as sellers, and UAVs as auctioneers. Last, to resist attack of malicious edge servers and ensure the task security, a novel trust evaluation method based on the comparison of true utility and expected utility is integrated in auction mechanism. The theoretical analysis and implementation results show that the proposed STMTO system not only achieve the best utility for devices and edge servers simultaneously, but also identify the malicious edge servers and protect task from attacks.

1. Introduction

Recently, the Internet of Things (IoT) has been developing rapidly. More and more smart sensing devices are deployed in demand to sense data [1], communicate and perform tasks collaboratively, thus promoting the development of IoT based applications [2,3]. According to IoT Analytics, the number of IoT devices worldwide is expected to reach 22 billion by 2025 [4]. Computing intensive task such as complex data process is beyond the computing capacity of IoT devices with simple hardware [5–7]. Therefore, offloading the tasks of IoT devices to the nearby edge servers with rich computing resource is an effective solution, which is also called edge computing [8–11]. There is much previous research of edge computing focusing on the scenario that IoT devices have direct access to the Internet, i.e. fully connected networks [12–14]. And some researchers study the task offloading under semi-connected networks, in which IoT devices roadside offload their tasks to

* Corresponding author.
E-mail address: xiongnaixue@gmail.com (N.N. Xiong).

https://doi.org/10.1016/j.ins.2021.05.020
0020-0255/

edge servers by mobile vehicles [15–17]. However, the research about task offloading under connectionless networks is lacking, such as the supervisory control and data acquisition (SCADA) system [18]. In such a system, many IoT devices spread over a wide area in order to monitor the specific process in a monitoring field [19,20]. This type of network has a broad prospect in many practical applications, especially in disaster, battlefield, smart city, health care and other scenarios [21–24].

Due to several tasks are beyond the computing capacity of IoT devices and the devices cannot access to the Internet directly [22,25,26], task offloading for such connectionless networks is a challenging issue which is studied in this paper [5,6,7]. Benefiting from the advance in mechanics, sensors and Artificial Intelligence (AI) [7,16,27], employing Unmanned Aerial Vehicles (UAVs) as relay to offload the tasks from IoT devices to edge severs is one of promising approaches [26,28,29]. The key to this approach is that the UAV is capable of 5G communication, so when the UAV flies into the communication range of IoT devices, it can act as a relay to establish communication with the Internet [20]. Therefore, it can offload the tasks to edge servers for computation, and also return the task results to devices, thus completing the task offloading process. For instance, Guo et al [30] proposed a UAV based task offloading strategy for such connectionless networks. In their proposed scheme, the UAV flies over the devices and receives tasks from devices. If the UAV is configured with rich computing resource, the UAV can process the tasks itself, or it offloads the tasks to a nearby edge server. Besides task offloading, UAVs can also be applied to collect sensing data form the IoT devices. The proposal of Jiang et al [31] is to use UAV as data mules to collecting data for each SCADA system. UAV has the advantages of high speed, independent of physical conditions such as terrain, on-demand dispatching and low cost, which makes it an ideal tool for task offloading in connectionless networks [30]. The current research has only achieved task offloading for connectionless networks, however, the deeper and more detailed research is still facing significant challenges, and some of such challenge issues are as follows.

(1) The security issue in task offloading. In the previous work of UAV based task offloading, edge servers were considered to be trusted, so it was believed that as long as the tasks and edge servers matched the satisfying results could be achieved. For example, Chen et al [32] applied multiple UAVs to collaboratively scout in post-disaster networks, where UAVs offload complex tasks to smart vehicles around the disaster area for execution in a double matching manner. However, many smart attack techniques have emerged for autonomous multi-robot systems, especially for UAVs. UAVs have relatively simple hardware due to cost considerations, so the adversary could compromise UAVs and control UAVs to perform a variety of malicious activities [10]. Similarly, task executors, i.e. smart vehicles in this scenario, are deployed by multiple parties and distributed over a wide area, which could also be compromised by adversary [14,33]. In this case, tasks offloaded to these malicious edge servers can cause serious consequences to the network. Therefore, it is urgent to develop defense approaches to protect the autonomous multi-robot systems against the attacks.

(2) The issue for considering the willingness of devices and edge servers in task offloading and how to maximize the utility of interaction between the two parties. Previously, the prerequisite of task offloading is generally that edge servers are always willing to provide service. Thus, the related studies only stand for the devices to maximize the performance of task offloading by matching the requirements of devices and the resource edge servers can provide, in which the willingness and utility of edge servers are not considered. However, this ideal mechanism often does not work in practice. The edge servers do not provide service for free and want to maximize their utility by task computation, while IoT devices desire to get satisfying resources and complete tasks with minimum cost [5]. Therefore, there is a game process between the two parties, which is a distributed interaction system [5]. In such interaction process, the bids for offloading tasks from devices and edge servers can hardly always satisfy the expected utility of each other. When the utility of devices and that of edge servers conflicts, the devices will change its offloading strategy to find another server that meets its expectation or adjust its expected utility. Therefore, in practice the task offloading is not always negotiated successfully between the two parties, and only considering the matching degree between the requirements of devices and the resources of edge servers does not always maximize the offloading utility [5]. Above all, the task offloading is a transaction process between devices and edge servers. It is significantly relevant to the system cost and affects the offloading decision in practical applications, which should not be ignored.

(3) The issue for secure and effective multi-UAV task offloading framework. Although some UAV based task offloading schemes have been proposed [30], most of them are for a specific issue, thus lacking a secure and effective multi-UAV task offloading framework, the overall performance of which is not high [34,35]. For example, there has been amount of research on UAV trajectory optimization, as well as some research on multi-UAV trajectory [30]. However, the trajectory optimization in previous studies seldom considered the density and distribution of IoT devices, which can affect the trajectory and the number of UAVs. And some task offloading research lacks the consideration of competition relationship between devices and edge servers, so it is not very applicable in practice. In addition, although there is a two-tier task offloading model based on the auction mechanism, the trust of edge servers is rarely considered in auction. Therefore, it is urgent to establish a secure and effective multi-UAV based task offloading system.

In this paper, a Smart and Trust Multi-UAV Task Offloading (STMTO) system is proposed to offload tasks in a collaborative way with multiple UAVs, where the task security can be ensured applying a new trust evaluation approach. The main contributions of our work are as follows:

(1) We propose a Smart and Trust Multi-UAV Task Offloading (STMTO) system to offload tasks from IoT devices to edge severs by UAVs. In STMTO system, a group of UAVs is dispatched to relay tasks from IoT devices to edge servers with strong computing capacity. A collaborative task collection scheme for multi-UAV based system is proposed to achieve the goal of minimizing energy consumption and the task processing delay.
(2) An effective many-to-many task double auction mechanism is established to reach maximum utility for devices and edge servers in connectionless networks. In this auction mechanism, devices act as buyers, edge servers as sellers, and UAVs as auctioneers. By the theoretical analysis, the proposed auction model can reach Bayesian-Nash Equilibrium, and the theoretical optimal bid strategy for buyers and sellers is given, which can maximize the utility of both parties simultaneously.
(3) We propose a novel trust evaluation method for edge servers, which is integrated into the auction mechanism. Based on the result of trust evaluation, trust based task double auction model can resist attack of malicious edge servers and ensure the task security. By comparing the true utility brought to devices after the tasks processed and the utility expected by devices before offloading the tasks, the trust value of the edge servers processing these tasks can be evaluated. In addition, a trust recommendation method is applied in the multi-UAV based system to extend the scope of trust evaluation. In this way, the UAVs can make offloading decisions comprehensively considering the bid for tasks and the trust value of edge servers.

The rest of this paper is organized as follows. In Section 2, the related works are reviewed. The system model and problem statement are presented in Section 3. In Section 4, the Smart and Trust Multi-UAV Task Offloading (STMTO) system is proposed. Then, Section 5 provides the experimental results. Conclusions are given in Section 6.

2. Related works

Recently, robots in UAV form have been widely used for data collection and task offloading in many IoT scenarios due to their high mobility and deployment flexibility [36]. In task offloading, UAVs are equipped with servers that can receive and process complex tasks from devices or continue to transfer tasks to edge servers or cloud for computation [30,31,36]. This edge computing paradigm solves the problem that tasks are beyond the computing capacity of devices for connectionless networks. There have been many studies on this task offloading mode. For example, Hu et al [36] designed a task offloading mechanism of orthogonal multiple access for UAVs to provide computational resource to devices, jointly optimizing the UAV trajectory and task offloading ratio to minimize the task processing delay. Xiong et al [37] also considered the UAV-enhanced edge computing scenario and proposed an algorithm to jointly optimize the UAV trajectory and the bit allocation in task transmission, which minimizes the system energy consumption while satisfying the task delay threshold. Zhang et al [38] established a three-tier UAV based edge computing system on the social vehicular network, which integrates computational offloading, content caching and other factors to solve the dynamic allocation of resource. Jiang et al [31] also applied UAVs as task offloaders in the SCADA System to collect tasks from devices and upload them to edge servers for execution. In addition, some research of task offloading focused on the utility relationship between task source and task executors. For instance, Misra et al [39] proposed a two-tier task offloading model based on an auction mechanism with the goal of optimizing the utility of device. Mashhadi et al [40] applied two neural networks to optimize the task offloading decision and the bidding strategy of devices, respectively.

Some task offloading strategies apply Single UAV (SU) to design an enhanced edge computing system, and the research focuses on optimizing the flight trajectory and task offloading ratio to reach the better performance of system. However, as the number of IoT devices increases and the distribution area expands [35], the energy consumption of single UAV increases excessively, and the Quality of Service (QoS) provided by the UAV can hardly be maintained at the level that it was when the device number was small [41]. Therefore, considering the requirements of networks with larger scale, some studies have extended the architecture of edge computing to a multi-UAV collaborative working model [26,35,41,42]. Kim et al [26] proposed a collaborative relay communication network for multiple UAVs. The scenario in their study is that IoT devices are remotely deployed. Due to the short communication distance, only one UAV acts as relay is still insufficient to make the IoT devices communicate with the base station. Therefore, Kim et al [26] used multiple UAVs to establish a communication link from IoT devices to the base station, so that IoT devices have access to the base station. The work of Kim et al [26] focuses on minimizing the energy consumption of UAVs through optimal multi-hop transmission. Moreover, Wu et al [35] used multi-UAVs as two roles. On the one hand, UAVs act as mobile wireless power transfer to supply energy towards the IoT devices. On the other hand, UAVs are the information collectors to collect data from IoT devices. The goals of the trajectory optimization method proposed in this paper are to balance the energy consumption among UAVs, minimize the communication delay and maximize the energy utilization under the constraints of the UAV energy capacity, which is very complex. In addition, Wang et al [41] applied Multi-UAV System (MUS) as mobile edge computing nodes to provide computing resource to users. In [41] a two-layer optimization model is proposed for jointly optimizing multi-UAV deployment and task scheduling, which achieves the adaptive number of UAVs and minimization of system energy consumption. Besides issues such as task scheduling, Yang et al [42] also proposed a differential evolution based UAV deployment algorithm that enables the load balance of the distributed system while ensuring the quality of service of IoT nodes. In the task offloading architectures above, the multi-UAV system acts as a task receiver and is responsible for providing computational resource to the IoT

devices. However, in [32], the multi-UAV system is applied to scout for disaster in the network after damage, in which the UAVs transmit complex tasks to vehicles or base stations around the disaster area for computation. Chen et al [32] not only designed the cooperative working scheme for multi-UAVs, but also considered the utility relationship between the task offloaders and the task receivers. Therefore, a distributed double matching algorithm for optimal task scheduling is proposed. In the matching process of UAVs and vehicles, both parties have initiative to select the objects they want to cooperate with to ensure the maximization of their benefits. However, as the system scales up, the number of vehicles involved in processing tasks increases rapidly. And because the vehicles are deployed by multiple parties, their trustworthiness cannot be determined. Malicious vehicles may disguise as normal vehicles to participate in task matching and end up reporting fake task results, thus affecting the efficiency of disaster reconnaissance. Chen et al [32] and the related work mentioned above rarely focus on the issue of trust and task security, which is also a significant and urgent issue to be studied in task offloading.

In task offloading, the trustworthiness of edge servers largely affects the efficiency of task offloading. Some untrustworthy edge servers do not perform tasks properly and submit fake processing results intentionally to get paid from the devices, and even some malicious edge servers may launch attacks to violate the private information of the devices [43–45]. It is also pointed out in [44] that trusted fog nodes should be selected as offloading targets when task offloading, and untrustworthy fog nodes will reduce the efficiency of task processing and harm the data security. In conventional data transmission, encryption is often used to ensure data security. However, it is not applicable in task offloading because the task executors process the tasks on the premise of knowing the task content, which is different from forwarding packets directly in data collection. Therefore, Al-khafajiy et al [44] proposed a trust evaluation mechanism based on the historical behavior of fog nodes. The interactions between fog nodes can generate trust values for each other, and fog nodes with higher trust value are more likely to be selected for task processing in subsequent stage. In this way, the system can identify malicious objects during task offloading and ensure the task security.

Al-khafajiy et al [44] performed trust evaluation by quantifying the Quality of Service (QoS) and Quality of data Protection (QoP) of fog nodes. The wired connections between fog nodes are convenient for monitoring related parameter (e.g., bandwidth, data transmitting rate). Besides, there are other studies on protecting data reliability and security by detecting QoS and other similar indicators. For instance, Dr. Xiong proposed a general method and a specific reliable detector [46], which are the core technologies in this field, and lay the foundation for the research of reliable detecting and protecting. However, the scenario in this paper is that the devices have no access to the Internet and needs UAVs as medium to offload tasks, so devices have no direct interaction with edge servers, which makes it not feasible to evaluate trust by monitoring QoS and QoP. To effectively identify malicious edge servers while ensuring the utility of devices and trusted edge servers, we propose a novel trust evaluation method integrated in task auction mechanism: the trust of edge servers processing tasks is judged based on the difference between the true utility brought to the devices after task processing and the expected utility estimated by the devices before task offloading, which is different from other previous work. And MUS is applied to act as task offloader, auctioneer and trust evaluator simultaneously in the offloading process.

3. System model and problem statements

3.1. System model

In this paper, a three-tier architecture for collaborative task offloading is proposed, as shown in Fig. 1, including device layer, UAV layer and edge server layer. (1) Device layer: There are a large number of mobile devices deployed for perceiving and processing environmental data for various applications, such as weather forecast and traffic monitoring, represented as the set $D = \{d_1, d_2 \cdots d_m\}$. Mobile devices rely on battery power and have energy limitations, thus offloading computationally intensive tasks to edge servers for processing to prolong the lifetime. However, due to the manufacturing cost, the devices only contain simple hardware and cannot connect directly to the Internet. The devices can first transmit tasks wirelessly via Bluetooth, WiFi, etc. to the UAV, and then the tasks are transferred to the edge server for execution. In addition, mobile devices join the task offloading system by registration so that the UAVs can know their location and other information to design task collection strategy and identify the mobile devices during flight. (2) UAV layer: Multiple UAVs are scheduled for collecting and offloading tasks of devices in a collaborative way, denoted by $U = \{u_1, u_2 \cdots u_n\}$. Each UAV is responsible for the tasks of devices in one area and the areas do not overlap. The UAVs have two types of communication: short-range wireless transmission for devices, and communicating with the servers via the Internet. (3) Edge server layer: Many optional edge servers with rich computational resource are deployed at the edge of the network, denoted by $S = \{s_1, s_2 \cdots s_o\}$. Edge servers receive and process tasks from UAVs. After tasks being finished, edge servers return the results to the corresponding devices via UAVs and receive a certain payment from devices for processing tasks.

3.2. Task offloading model

Assume that in each round of task offloading, for any device d_i there is a task list $T_i = \{t_i^1, t_i^2 \cdots t_i^l\}$ that need to be offloaded to edge servers. Each task $t_i^j = \left(\alpha_i^j, c_i^j[k], \gamma_i^j, \delta_i^j\right)$ contains several attributes, where α_i^j is the data amount of t_i^j, γ_i^j is the iteration times needed to process t_i^j, and $c_i^j[k]$ is the computation amount for the k-th iteration. The complete process

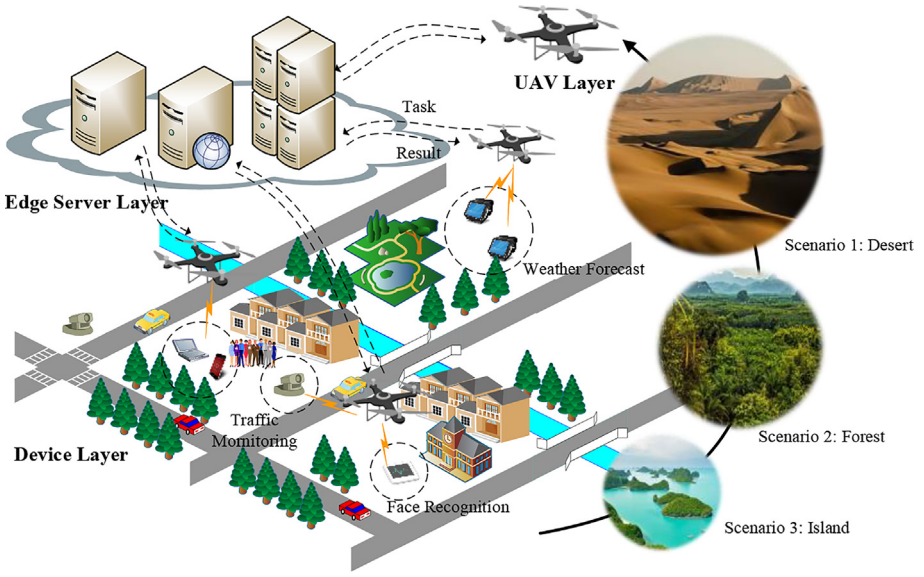


Fig. 1. Three-layer system model.

of offloading task t_i^j includes: d_i waiting for the UAV collecting t_i^j, d_i offloading t_i^j to the UAV, the UAV selecting the edge server s_w to forward t_i^j, and s_w processing t_i^j and returning the result. In the following we calculate the delay and energy consumption in each stage of the offloading process.

(1) Offloading Delay

In Eq. (1), D_w^{ij} is the delay that task t_i^j waits for the UAV, where τ_o^{ij} is the time of the UAV starting to offload t_i^j and τ_s^{ij} is the time that t_i^j is ready for offloading. D_{du}^{ij} in Eq. (2) is the delay for d_i to transmit t_i^j to the UAV, where W_{du} is the bandwidth of the transmission channel between the device and the UAV, P_i^s is the average power of signal transmission for d_i, and N_{du} is the Gaussian noise power in the channel. Similarly, the delay of the UAV transmitting t_i^j to the edge server is shown in Eq. (3). W_{us} is the bandwidth of the transmission channel between the UAV and the edge server, P_u^s is the signal transmission power of UAV, and N_{us} is the Gaussian noise power in the channel. In Eq. (4), D_p^{ijw} is the duration of the edge server s_w processing the task, where $\sum_{k=1}^{\gamma_i^j} c_i^j[k]$ is the total computation amount of task t_i^j and f_w is the CPU working frequency of s_w. Since the data amount of task result is small, its return time is ignored [47].

$$D_w^{ij} = \tau_o^{ij} - \tau_s^{ij}. \tag{1}$$

$$D_{du}^{ij} = \frac{\alpha_i^j}{W_{du}\log_2\left(1+\frac{P_i^s}{N_{du}}\right)}. \tag{2}$$

$$D_{us}^{ij} = \frac{\alpha_i^j}{W_{us}\log_2\left(1+\frac{P_u^s}{N_{us}}\right)}. \tag{3}$$

$$D_p^{ijw} = \frac{\sum_{k=1}^{\gamma_i^j} c_i^j[k]}{f_w}. \tag{4}$$

(2) Offloading Energy Consumption

E_{du}^{ij} is the energy consumption of d_i transmitting t_i^j to the UAV, where P_i^s is the sending power of d_i and P_u^r is the receiving power of UAV. Similarly, the energy consumption of offloading t_i^j from the UAV to the edge server is E_{us}^{ij}, shown in Eq. (6),

where P_u^s is the average sending power of the UAV and P_w^r is the receiving power of s_w. In Eq. (7), E_p^{ijw} is the energy consumption of s_w to process t_i^j and P_w^p is the average power of s_w.

$$E_{du}^{ij} = \frac{\alpha_i^j}{W_{du}\log_2\left(1+\frac{P_i^s}{N_{du}}\right)}\left(P_i^s + P_u^r\right). \tag{5}$$

$$E_{us}^{ij} = \frac{\alpha_i^j}{W_{us}\log_2\left(1+\frac{P_u^s}{N_{us}}\right)}\left(P_u^s + P_w^r\right). \tag{6}$$

$$E_p^{ijw} = \frac{\sum_{k=1}^{\gamma_i^j} c_i^j[k]}{f_w} P_w^p. \tag{7}$$

3.3. Task auction model

Devices offload tasks to edge servers for execution and need to pay certain reward to them. The edge servers obtain utility for processing tasks, while the tasks processed successfully also bring benefit to devices. Devices have multiple edge servers to select when offloading tasks, and edge servers can also choose to receive and process tasks from different devices for own utility. Therefore, a many-to-many task auction model is established for multiple devices and edge servers, where devices act as buyers, edge servers as sellers, and UAVs as auctioneers. In the following we formulate the expected utility of offloading task t_i^j to s_w for devices and servers. Device d_i has a public bid v_d^{ij} for t_i^j and a private valuation $\widehat{v}_d^{ij}$. v_d^{ij} is the price d_i expects to pay to the edge server for processing t_i^j, and $\widehat{v}_d^{ij}$ is predicted value of t_i^j brought to d_i when t_i^j is finished. We assume that d_i is willing to offload t_i^j at a lower cost, so $\widehat{v}_d^{ij} > v_d^{ij}$. Similarly, each edge server s_w also has a public bid z_w^{ij} and a private valuation $\widehat{z}_w^{ij}$ for task t_i^j. z_w^{ij} is the price set by s_w for processing task t_i^j. And $\widehat{z}_w^{ij}$ is the cost of processing t_i^j, which is positively correlated with the computation amount and memory required, as shown in Eq. (8). c_w^{unit} is the unit processing cost of s_w, and λ_α and λ_c are the weight of memory and computation of the task, respectively. In our auction model, we assume that the transaction price is defined as $p^{ij} = \frac{v_d^{ij}+z_w^{ij}}{2}$, i.e. the middle value of public bids from device and edge server. In other word, the device d_i should pay for s_w the payment of p^{ij} when both parties are willing to process task offloading. Therefore, the expected utility U_d^{ij} for d_i is given in Eq. (9), where the expected utility is the difference between the transaction price and private valuation if t_i^j is successfully offloaded, i.e. $o_d^{ij} = 1$, and 0 otherwise. Similarly, if the edge server s_w receives t_i^j, i.e. $o_s^{wij} = 1$, its expected utility is as shown in Eq. (10).

$$\widehat{z}_w^{ij} = c_w^{unit} \cdot \left(\lambda_\alpha \alpha_i^j + \lambda_c \sum_{k=1}^{\gamma_i^j} c_i^j[k]\right). \tag{8}$$

$$U_d^{ij} = \begin{cases} \widehat{v}_d^{ij} - \frac{v_d^{ij}+z_w^{ij}}{2}, o_d^{ij} = 1; \\ 0, o_d^{ij} = 0. \end{cases} \tag{9}$$

$$U_s^{wij} = \begin{cases} \frac{v_d^{ij}+z_w^{ij}}{2} - \widehat{z}_w^{ij}, o_s^{wij} = 1; \\ 0, o_s^{wij} = 0. \end{cases} \tag{10}$$

3.4. Problem statement

Due to advantages such as high mobility and low cost, UAV has become a common tool for data collection and task offloading for devices that have no direct access to the Internet. However, when there are a larger number of devices deployed over a wide area, single UAV needs to traverse the devices with excessively long flight distance, which increases the energy consumption and the workload of single UAV. In addition, the collection time of tasks increases, resulting in a higher waiting delay and some tasks cannot be guaranteed to be processed before deadline. To solve these problems, Multi-UAV System (MUS) is applied to offload tasks in a collaborative way. In order to ensure the load uniformity on MUS, while reducing the offloading cost and task processing latency, designing an effective MUS collaborative task offloading strategy becomes an urgent issue.

Moreover, as edge servers are deployed by multiple parties, their trustworthiness cannot be determined. There may be untrustworthy edge servers participating in the task auction hosted by UAVs, disrupting the task auction process by making malicious bids, or discarding tasks after receiving rewards by transaction with devices, resulting in a lower task processing success rate. Therefore, it is important to establish an auction mechanism that incorporates trust evaluation for identifying malicious servers to ensure the utility of devices and trusted servers and the security of tasks.

The main idea of this paper is to design a MUS enhanced task offloading system based on the problems above. There are four main goals of the system, shown as Eqs. (8)–(11). (1) Minimize the energy consumption of offloading E, including the flight cost of UAVs E_f, the task transmission cost between devices and UAVs E_{du} and that between UAVs and edge servers E_{us}, and the processing energy consumption of edge servers E_p. (2) Minimize the task processing delay D, including the device waiting delay D_w, the communication delay between devices and UAVs D_{du} and that between UAVs and edge servers D_{us}, and the task processing delay D_p. (3) Maximize the utility U during task auction, including the total utility U_d of devices and the total utility U_s of edge servers. (4) Maximize the task processing success rate R, where N_s is the number of tasks whose processing results reach the expected valuation of the device, and N_t is the total number of tasks.

$$\min(E) = \min\left(E_f + E_{du} + E_{us} + E_p\right). \tag{11}$$

$$\min(D) = \min\left(D_w + D_{du} + D_{us} + D_p\right). \tag{12}$$

$$\max(U) = \max(U_d + U_s). \tag{13}$$

$$\max(R) = \max\left(\frac{N_s}{N_t}\right). \tag{14}$$

Equation (11) has E_{du} and E_{us} as constant values for a constant number of tasks and device transmission power, while E_f and E_p varies with the number of UAVs as well as the task allocation strategy. To simplify the problem, we consider the power and speed of UAVs to be constant during flight, and thus Eq. (15) shows the calculation of E_f, where P_f is the flight power of UAVs, L_i is the trajectory length of the UAV u_i, and v_f is the flight speed of UAVs.

$$E_f = P_f \frac{\sum_{u_i \in U} L_i}{v_f}. \tag{15}$$

Similarly, in Eq. (12), D_{du} and D_{us} are constant for a constant number of tasks and device transmission power, and D_w varies with the task collection strategy: the smaller the number of devices traversed by one UAV, the shorter the offloading time for a round, which brings smaller waiting delay. In addition, the processing time of different edge servers for one task varies, so D_p is affected by the offloading decision of the UAV. Therefore, the optimization goals can be re-defined as Eq. (16).

$$\begin{cases} \min(E) = \min\left(E_f + E_p\right), \\ \min(D) = \min\left(D_w + D_p\right), \\ \max(U) = \max(U_d + U_s), \\ \max(R) = \max\left(\frac{N_s}{N_t}\right). \end{cases} \tag{16}$$

4. Our proposed STMTO system

4.1. The design of collaborative task collection scheme

In STMTO, n UAVs are scheduled to work collaboratively, and each UAV is responsible for task offloading of devices in an area. To reduce the flight energy consumption, it is necessary to divide working area for each UAV and design the flight trajectory before traversing the devices and collecting tasks. Firstly, the device layer is divided into n areas, which constitute the set $A = \{a_1, a_2 \cdots a_n\}$. Each area contains some devices, and the UAV u_i is responsible for offloading the tasks of devices in a_i. Moreover, the UAVs should traverse all devices in one round and there is no overlap between two working areas, i.e., $\forall a_i \bigcap a_j = \varnothing, \bigcup_{i=1}^{n} a_i = D$. Devices with similar physical locations are suitable to be divided into the same area, so that the UAV traversing the devices requires less flight time and the waiting delay of tasks can be reduced, while saving the flight energy. Therefore, a Location Based Working Area Division Algorithm based on the thought of K-Means clustering algorithm is designed, which is the first stage of collaborative task collection scheme. K-Means is an unsupervised learning iterative algorithm, which is mainly used to deal with sample classification problems with known category number.

In the device layer, each device d_i is deployed by the system, so the coordinate $p_i = (x_i, y_i)$ of d_i are known. Firstly, the working area center $c_j = \left(x_j^c, y_j^c\right)$ is randomly initialized for each UAV u_j. L_{ij} is defined to express the distance between device d_i and area center c_j, and then d_i selects the area center with the smallest distance and joins it. After that each area center c_j is relocated according to Eq. (17), i.e., the updated coordinate $c_j^{'}$ is the average value of coordinate of all devices in a_j. Repeat the

steps above to calculate L_{ij} and update the area d_i joining and the coordinate of c_j until c_j does not change. Algorithm 1 shows the main process, where C_{pre} is the set of area centers in the previous iteration, j^* is the label of area d_i joining, and j^*_{pre} is the label of area d_i joining in the previous iteration.

$$c_j^{'} = \frac{\sum_{d_i \in a_j}(x_i, y_i)}{|a_j|}. \tag{17}$$

Working area division allows each UAV to be responsible for some of the devices in the smallest possible range, which reduces the flight energy and balances the load of system. To further reduce the flight distance of UAVs, we need to design algorithms to find the shortest flight trajectory. Here the Multi-UAV Trajectory Optimization Algorithm based on the thought of Genetic Algorithm is proposed to optimize the trajectory for MUS. Genetic Algorithm is a method that searches for sub-optimal solutions by simulating the natural evolutionary process. In solving complex combination problems, Genetic Algorithm can obtain optimization results faster compared to other conventional optimization algorithms. The following are the main steps of the Trajectory Optimization Algorithm.

Algorithm 1 Location Based Working Area Division Algorithm

Input: $P = \{p_1, p_2 \cdots p_m\}$, n
Output: $A = \{a_1, a_2 \cdots a_n\}$
Randomly initialize $C = \{c_1, c_2 \cdots c_n\}$
While $C! = C_{pre}$:
  For p_i **in** P:
    For c_j **in** C:
      $L_{ij} \leftarrow ||p_i - c_j||$
    End for
    $j^* \leftarrow argmin(L_{ij})$
    If $j^*! = j^*_{pre}$:
      Remove d_i from previous area
      $a_{j^*} \leftarrow a_{j^*} \bigcup d_i$
    End if
  End for
  Update C according to Eq. (1)
End While
Return A

1) Firstly, we specify the optimization objective. The UAV u_j is responsible for the task offloading in the area $a_j = \{d_i | d_i \in D\}$. The flight distance L_f changes with u_j choosing different orders to traverse the devices in a_j. Assume that a_j contains four devices, and then there is one of the traversing orders, i.e. trajectories, $g = (d_1, d_2, d_3, d_4)$, whose corresponding flight distance $L_f(g)$ is shown in Eq. (18). Therefore, our goal is to find a trajectory such that L_f reaches a minimum value. The idea of searching the optimal trajectory is as follows. Firstly, randomly generate a number of trajectories $G = \{g_1, g_2 \cdots g_x\}$. Secondly, perform transformations on the trajectories, such as Step 2 and Step 3, and filter out the part of trajectories with smaller $L_f(g)$ into the next iteration. After several iterations, the optimal trajectory in G is output. The initial trajectory set for each UAV constitutes $\mathbf{G} = \{G_1, G_2 \cdots G_n\}$.

$$L_f(g) = \sum_{i=1}^{3} ||p_i - p_{i+1}|| + ||p_4 - p_1||. \tag{18}$$

2) Exchange of two orders. For the UAV u_j, choose any two trajectories as pa_a and pa_b from G_j, and exchange the partial device order of them to generate two new trajectories ch_a and ch_b. The exchange process is as follows. Firstly,pa_a selects the breakpoint to divide the device order into two segments. Then eliminate the devices appearing in the former segment of pa_a from pa_b, and the remaining devices are used as the former segment of the new trajectory ch_a, and the former segment

Algorithm 2 Multi-UAV Trajectory Optimization Algorithm

Input: $P = \{p_1, p_2 \cdots p_m\}$, n,$r = 0$
Output: $\mathbf{G} = \{G_1, G_2 \cdots G_n\}$
$A \leftarrow$Execute Algorithm 1
Initialize $\mathbf{G} = \{G_1, G_2 \cdots G_n\}$ //Step 1
While $r < R$:
  For G_i **in G**:
    For pa_a **and** pa_b **in** G_i: //Step 2
      $ch_a, ch_b \leftarrow Exchange(pa_a, pa_b)$
      $G_i \leftarrow G_i \bigcup ch_a, ch_b$
    End for
    $Sort(G_i)$
    For g **in** G_i: //Step 3
      Calculate $L_f(g)$
      If $L_f(g) > L_f^{aver}$ **and** $random(0,1) < \psi$:
        $g \leftarrow G_i[0]$
        $g \leftarrow Swap(g)$
      End if
    End for
    Remove half of G_i with larger L_f //Step 4
    $G_i \leftarrow shuffle(G_i)$
  End for
  $r++$
End while
ReturnG

of pa_a is used as the latter segment of ch_a. Similarly, another new trajectory ch_b can be generated by dividing pa_b. After each pair of orders in G_j is exchanged, the capacity of G_j will be doubled.

3) Copy and swap. After the transformation in Step 2, $L_f(g)$ is calculated for each trajectory in G_j. The trajectories with larger $L_f(g)$ than the average value L_f^{aver} in G_j are identified as poor solutions, so we continue to transform them to reduce $L_f(g)$. First, we randomly select part of these poor trajectories at a certain possibility ψ, and replace them with the same number of best-performing trajectories, so as to improve the general performance of G_j. Secondly, to ensure the diversity of trajectories in G_j, each trajectorie replaced randomly chooses four breakpoints $0 < p_a < p_b \leq p_c < p_d < |a_j|$, dividing itself into five segments. Finally, The second and third segments, i.e., seg_{ab} and seg_{cd}, swap the positions in the order to complete the transformation.

4) After Step 2 and Step 3, the number of trajectories in G_j is doubled. To maintain the capacity of G_j, half of the trajectories with smaller $L_f(g)$ are filtered out as the input in the next round of optimization. The optimized trajectory can be obtained by repeating the steps above. Algorithm 2 shows the process of trajectory optimization, where R is the number

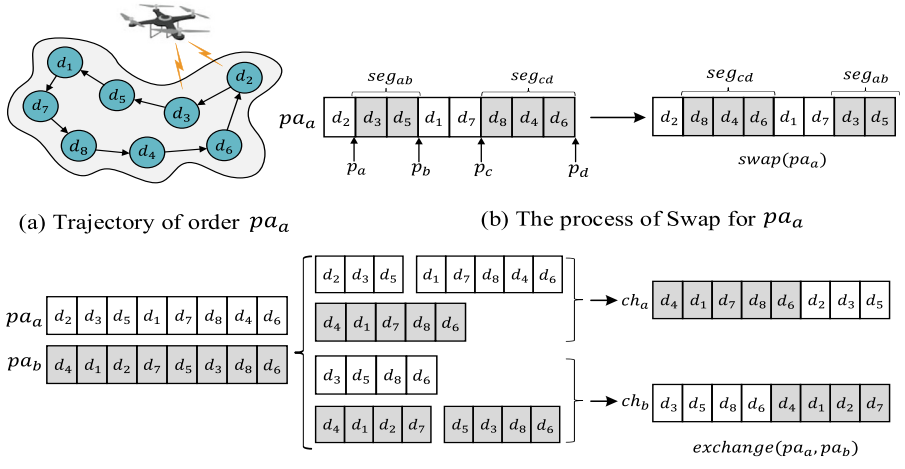


Fig. 2. Examples of order transformation in Step 2 and Step 3.

of iterations, and *Exchange* and *Swap* are the transformation functions for trajectories in Step 2 and Step 3. Fig. 2 gives the examples of *Exchange* and *Swap*.

The above cooperative collection strategy is designed on the basis of the uniform distribution of devices, so the number of devices in each working area and the flight distance of UAVs do not vary much. However, for the uneven distribution of devices, the above strategy based on device density may lead to uneven loading of UAVs: some UAVs are responsible for area where the devices are sparsely distributed, and the time to complete a round of offloading is relatively short. However, some UAVs are responsible for areas with dense distribution of devices, so the offloading duration is much longer, and the waiting time D_w of the devices at the end of the traversal sequence is longer, which is not conducive to the timeliness of task processing. Therefore, for the above special case, a dynamic trajectory sharing strategy for MUS should be proposed to further balance the offloading cost of MUS.

$$max(e_j) = max(\lambda_c \cdot c_r^j - \lambda_L \cdot \Delta L_j).u_j \in U, u_j \neq u_i \tag{19}$$

For any UAV u_i, after completing the planned device traversal, it remotely queries the completion status of other UAVs via the 5G. If the other UAVs have not completed, u_i selects the UAV u_j that needs help by considering the incremental flight distance ΔL to another working area and the number of devices to be traversed c_r. Eq. (19) expresses the selection goal of u_i, where e_j is the combined evaluation value of u_i to other UAVs u_j, λ_c and λ_L represent the weights of the two considerations, respectively. From Eq. (19), it can be seen that u_i prefers the UAV that is closer to itself and has a higher number of devices with uncompleted task auction. After u_i selects a suitable u_j, u_j sends its device traversal list to u_i. u_i immediately flies to the last device in the list and performs task collection from back to front. In this way u_i and u_j work in parallel in opposite directions, and the task collection of all devices is completed when the two UAVs meet. In this way, UAVs with light load can dynamically help the partners collect their tasks after finishing own job, which better reflects the concept of collaborative work and proves the scalability and applicability of the proposed strategy. In Section 5, we will analyze the performance of collection strategy in detail.

4.2. The design of trust based task auction mechanism

In task offloading, UAVs act as auctioneers to collect bids from devices and edge servers and make offloading decisions based on the bids. In this subsection we focus on two problems in the auction process mentioned in Problem Statement to design a trust based double auction mechanism. (1) The devices and edge servers have some conflict of utility. The devices want to pay less price to the servers to reduce the offloading cost and improve its utility. However, when the bids of devices are too low to meet the expectation of servers, the task auction fails. Secondly, there may be participants who provide malicious bids to impact the task auction. Therefore, it is necessary to set a reasonable bid range in the auction mechanism and analyze the optimal bidding strategy for devices and edge servers according to the bidding laws of both parties, which improves the auction success rate and maximize the utility. (2) Untrustworthy edge servers may directly discard tasks after receiving them from UAVs, thus affecting the task processing efficiency. To avoid selecting untrustworthy edge servers, trust evaluation is integrated in the auction mechanism to monitor the task processing quality of edge servers, so as to screen trusted edge servers for task processing and ensure the security of tasks.

The design of auction mechanism is preceded by the following assumptions.

Assumption 1:. The private valuation of tasks by both devices and edge servers is confidential, which is unknown to the other participants. However, the probability distribution function of private valuation between devices and edge servers is consensus.

Assumption 2:. To avoid malicious bids from some participants that interfere with the auction process, UAVs review all bids and only allows bids within predefined bidding range to be valid.

Assumption 3:. All participants are risk-neutral, i.e., the device expects to pay a lower price for its own utility, but too low a bid will result in no edge server receiving the task, so the device decides the final bid considering both the auction success rate and its own utility. The same is true for edge servers.

The double auction model proposed above meets the basic characteristics of the static game with incomplete information: (1) The two parties do not know the private valuation and public bids of each other. (2) The decisions of the two parties do not depend on the behavior of the other party, so there is a Bayesian-Nash Equilibrium, and its corresponding bidding strategy is the optimal bidding strategy. In the following, we conduct a theoretical analysis for the optimal bidding strategy for devices and edge servers.

First, we specify the optimization objective of the optimal bidding strategy, i.e., maximizing the expected utility of devices and edge servers, as shown in Eq. (20).

$$Objective: \max\left(E\left(U_d^{ij}\right)\right), \max\left(E\left(U_s^{wij}\right)\right). \tag{20}$$

In the auction, participants determine the public bids based on their own private valuation for task t_i^j. To ensure that the bids are within a reasonable range like mentioned in Assumption 2, we first set the upper and lower valuation bound b_{min}^{ij} and b_{max}^{ij} for each task t_i^j, which is related to the task information of t_i^j, calculated in Eqs. (21) and (22). θ_{min} and θ_{max} are constant value, derived from the statistical analysis of historical private valuation of devices and edge servers. The private valuation of devices and edge servers for t_i^j must satisfy $\hat{v}_d^{ij} \in \left[b_{min}^{ij}, b_{max}^{ij}\right]$ and $\hat{z}_w^{ij} \in \left[b_{min}^{ij}, b_{max}^{ij}\right]$.

$$b_{min}^{ij} = \theta_{min}\left(\lambda_\alpha \alpha_i^j + \lambda_c \sum_{k=1}^{\gamma_i^j} c_i^j[k]\right). \tag{21}$$

$$b_{max}^{ij} = \theta_{max}\left(\lambda_\alpha \alpha_i^j + \lambda_c \sum_{k=1}^{\gamma_i^j} c_i^j[k]\right). \tag{22}$$

To simplify the problem, we consider that the bids and the private valuation are positively correlated, so the bids v_d^{ij} and z_w^{ij} of devices and servers are expanded to functions $v_d^{ij}\left(\hat{v}_d^{ij}\right)$ and $z_w^{ij}\left(\hat{z}_w^{ij}\right)$ respectively, as shown in Eqs. (23) and (24). Therefore, according to the definition domain $\hat{v}_d^{ij}, \hat{z}_w^{ij} \in \left[b_{min}^{ij}, b_{max}^{ij}\right]$, the value domain of $v_d^{ij}\left(\hat{v}_d^{ij}\right)$ and $z_w^{ij}\left(\hat{z}_w^{ij}\right)$ can be redefined: $v_d^{ij}\left(\hat{v}_d^{ij}\right) \in \left[k_d^{ij} \cdot b_{min}^{ij} + \varepsilon_d^{ij}, k_d^{ij} \cdot b_{max}^{ij} + \varepsilon_d^{ij}\right]$, $z_w^{ij}\left(\hat{z}_w^{ij}\right) \in \left[k_w^{ij} \cdot b_{min}^{ij} + \varepsilon_w^{ij}, k_w^{ij} \cdot b_{max}^{ij} + \varepsilon_w^{ij}\right]$.

$$v_d^{ij}\left(\hat{v}_d^{ij}\right) = k_d^{ij} \cdot \hat{v}_d^{ij} + \varepsilon_d^{ij}. \tag{23}$$

$$z_w^{ij}\left(\hat{z}_w^{ij}\right) = k_w^{ij} \cdot \hat{z}_w^{ij} + \varepsilon_w^{ij}. \tag{24}$$

Assume that the bids of devices and edge servers satisfy the uniform distribution within their bidding range respectively, thus their probability density functions are shown in Eqs. (25) and (26).

$$f\left(v_d^{ij}\right) = \begin{cases} \frac{1}{k_d^{ij}\cdot\left(b_{max}^{ij} - b_{min}^{ij}\right)}, v_d^{ij} \in \left[k_d^{ij} \cdot b_{min}^{ij} + \varepsilon_d^{ij}, k_d^{ij} \cdot b_{max}^{ij} + \varepsilon_d^{ij}\right], \\ 0, else. \end{cases} \tag{25}$$

$$f\left(z_w^{ij}\right) = \begin{cases} \frac{1}{k_w^{ij}\cdot\left(b_{max}^{ij} - b_{min}^{ij}\right)}, z_w^{ij} \in \left[k_w^{ij} \cdot b_{min}^{ij} + \varepsilon_w^{ij}, k_w^{ij} \cdot b_{max}^{ij} + \varepsilon_w^{ij}\right], \\ 0, else. \end{cases} \tag{26}$$

It is generally stipulated in the auction mechanism that a transaction occurs when the bid of buyer is higher than that of the seller. Thus, when $v_d^{ij} > z_w^{ij}$, the UAV offloads t_i^j to s_w, while the true price $p^{ij} = \frac{v_d^{ij} + z_w^{ij}}{2}$ should be paid by d_i. Now we stand for d_i and s_w respectively, fixing their bids and calculating the probability of success transaction as well as the expected utility for both parties. First, v_d^{ij} is fixed and z_w^{ij} satisfies the probability distribution function $f\left(z_w^{ij}\right)$, from which the expected utility $E\left(U_d^{ij}\right)$ of d_i offloading t_i^j can be obtained, as shown in Eq. (27). Similarly, the expected utility of s_w about t_i^j can be calculated in Eq. (28).

$$E\left(U_d^{ij}\right) = \int_{k_w^{ij}\cdot b_{min}^{ij} + \varepsilon_w^{ij}}^{v_d^{ij}} \left(\hat{v}_d^{ij} - \frac{v_d^{ij} + z_w^{ij}}{2}\right) df\left(z_w^{ij}\right). \tag{27}$$

$$E\left(U_s^{wij}\right) = \int_{z_w^{ij}}^{k_d^{ij}\cdot b_{max}^{ij} + \varepsilon_d^{ij}} \left(\frac{v_d^{ij} + z_w^{ij}}{2} - \hat{z}_w^{ij}\right) df\left(v_d^{ij}\right). \tag{28}$$

Integrating the two formulas above, we get Eqs. (29) and (30). It can be seen that $E\left(U_d^{ij}\right)$ and $E\left(U_s^{wij}\right)$ are quadratic functions on v_d^{ij} and z_w^{ij} respectively, so the optimization problem is convex and there are maximum for $E\left(U_d^{ij}\right)$ and $E\left(U_s^{wij}\right)$. The independent variables when the function value reaches maximum are the optimal bidding strategies for d_i and s_w, $v_d^{ij^*}$ and $z_w^{ij^*}$.

$$E\left(U_d^{ij}\right) = \frac{v_d^{ij} - \left(k_w^{ij} \cdot b_{min}^{ij} + \varepsilon_w^{ij}\right)}{k_w^{ij} \cdot \left(b_{max}^{ij} - b_{min}^{ij}\right)} \left(\hat{v}_d^{ij} - \frac{v_d^{ij}}{2} - \frac{k_w^{ij} \cdot b_{min}^{ij} + \varepsilon_w^{ij} + v_d^{ij}}{4}\right). \tag{29}$$

$$E\left(U_s^{wij}\right) = \frac{k_d^{ij} \cdot b_{max}^{ij} + \varepsilon_d^{ij} - z_w^{ij}}{k_d^{ij} \cdot \left(b_{max}^{ij} - b_{min}^{ij}\right)} \left(\frac{z_w^{ij}}{2} + \frac{z_w^{ij} + k_d^{ij} \cdot b_{max}^{ij} + \varepsilon_d^{ij}}{4} - \hat{z}_w^{ij}\right). \tag{30}$$

Let $\frac{dE\left(U_d^{ij}\right)}{dv_d^{ij}} = 0$ and $\frac{dE\left(U_s^{wij}\right)}{dz_w^{ij}} = 0$, and combine Eqs. (23) and (24), and then the optimal bid strategy are obtained as Eqs. (31) and (32).

$$v_d^{ij*} = \frac{8\widehat{v}_d^{ij} + 3b_{min}^{ij} + b_{max}^{ij}}{12}. \tag{31}$$

$$z_w^{ij*} = \frac{8\widehat{z}_w^{ij} + b_{min}^{ij} + 3b_{max}^{ij}}{12}. \tag{32}$$

Algorithm 3 illustrates the process of task double auction. The UAV u_j collects the public bids and the task information when it arrives over d_i and uploads the task information to each edge server. When collecting the bids of devices, u_j firstly check whether the bids are greater than the tolerable minimum, in which π_v is the tolerable parameter for devices. If not, the bids should be return back for modification. Then the edge servers estimate the processing cost and provide public bids based on the task information. Similarly, u_j will check whether the bids are smaller than the tolerable maximum, in which π_w is the tolerable parameter for edge servers. Unqualified bids will be filtered out. To incentivize the bidding speed of edge servers and reduce the auction duration, u_j matches the bids of both parties by the bidding time order of edge servers. The edge servers that bid early have greater priority. The auction opportunity is set twice for each task, and the device d_i has the opportunity to modify the public bids for the tasks that are left in the first round. In the following, we prove the rationality of the auction process in terms of individual rationality and incentive compatibility.

Algorithm 3 Task Double Auction Process

For d_i **in** a_j:
  For t_j^i **in** T_i:
    d_i compute $\widehat{v}_d^{ij}$ and v_d^{ij}
    u_j collects t_j^i and v_d^{ij}
    If $v_d^{ij} < \pi_v \cdot b_{min}^{ij}$:
      u_j notify d_i to modify the bid of t_j^i
    End if
  End for
  u_j uploads task information from d_i
  For s_w **in** S:
    s_w computes $\widehat{z}_w^{ij}$ and z_w^{ij} for each task in T_i
    s_w sends public bids to u_j
    If $z_w^{ij} > \pi_w \cdot b_{max}^{ij}$:
      u_j filter z_w^{ij} out
    End if
  End for
  For t_j^i **in** T_i:
    $Z^{ij} \leftarrow Sort\left(z_w^{ij}\right)$ by bidding time
    For z^{ij} **in** Z^{ij}:
      If $z^{ij} < v_d^{ij}$:
        u_j uploads t_j^i to the corresponding server
        Break
      End if
    End for
  For t_j^i **in** the rest tasks:
    d_i modify $v_d^{ij} \rightarrow v_d^{ij} + \varepsilon$ and bid again
    Repeat auction process
      If no $z^{ij} < v_d^{ij} + \varepsilon$:
        Auction fails
      End if
    End for
End for

Theorem 1:. *If devices and edge servers follow the optimal bidding strategy analyzed above, the auction process reaches individual rationality.*

Proof:. *If* d_i *and* s_w *are successfully matched about* t_j^i, *we have* $z_w^{ij} < v_d^{ij}$. *The expected true utility of device* d_i *is* $E\left(U'^{ij}_d\right) = E\left(\widehat{v}_d^{ij} - p^{ij}\right)$. *From the above equation we can get*:

$$E\left(\widehat{v}_d^{ij} - p^{ij}\right) = E\left(\widehat{v}_d^{ij} - \frac{v_d^{ij^*} + z_w^{ij^*}}{2}\right) = E\left(\widehat{v}_d^{ij} - \frac{8\widehat{v}_d^{ij} + 8\widehat{z}_w^{ij} + 4b_{min}^{ij} + 4b_{max}^{ij}}{12}\right)$$

$$= E\left(\frac{2}{3}\widehat{v}_d^{ij} - \frac{1}{3}\widehat{z}_w^{ij} - \frac{1}{6}\left(b_{min}^{ij} + b_{max}^{ij}\right)\right)$$

$$> E\left(\frac{1}{3}\widehat{v}_d^{ij}\right) - \frac{1}{6}\left(b_{min}^{ij} + b_{max}^{ij}\right) = 0. \tag{33}$$

$E\left(U'^{ij}_d\right)$ is positive in this case, so for d_i is individually rational. Similarly, it can be proved that $E\left(U'^{wij}_s\right) = E\left(p^{ij} - \widehat{z}_w^{ij}\right) > 0$ for s_w. (2) If no server is willing to receive t_j^i or s_w bids fail, we have $E\left(U'^{ij}_d\right) = 0$ and $E\left(U'^{wij}_s\right) = 0$, respectively. In summary, the above auction process for any devices and servers has $E\left(U'^{ij}_d\right) \geq 0$ and $E\left(U'^{wij}_s\right) \geq 0$, satisfying the individual rationality. □

Theorem 2:. *If devices and edge servers follow the optimal bidding strategy analyzed above, the auction process reaches incentive compatibility.*

Proof:. The criteria of the UAVs for screening edge servers are the bidding time and the bidding value. Provided that the server knows the screening criteria, if s_w is willing to bid t_j^i, then s_w will calculate the public bid and transmit it to the UAV in the shortest possible time to get the top position in the bidding list. In the meanwhile, this mechanism is consistent with the overall goal of the system: to shorten the auction duration and reduce the waiting delay of tasks, which satisfies incentive compatibility. Secondly, for own utility, s_w may deviate from the optimal bid $z_w^{ij^*}$ to $z_w^{ij^*} + \varepsilon$, but this deviation will reduce the probability of successful transaction and lead the true utility of s_w to 0. And it can be seen in the analysis above that $z_w^{ij^*}$ satisfies the maximum expected utility of s_w when its bidding is smaller than that of d_i, so $E\left(U_s^{wij}\left(z_w^{ij^*}\right)\right) > E\left(U_s^{wij}\left(z_w^{ij^*} + \varepsilon\right)\right)$. If the bids of the servers all follow the optimal offer strategy, the overall expected utility of servers U_s are also maximized, as shown in Eq. (34). Therefore, the goal of the individual seller is consistent with the goal of system.

$$\max(E(U_s)) = \sum\nolimits_{s_w \in S} \sum\nolimits_{d_i \in D} \sum\nolimits_{t_i^j \in T_i} E\left(U_s^{wij}\left(z_w^{ij^*}\right)\right) o_s^{wij}. \tag{34}$$

□

The game model established above yields a generalized solution of Bayesian-Nash equilibrium from the theoretical analysis, where devices and servers following the optimal bidding strategy can maximize the utility of both parties simultaneously. Moreover, the trustworthiness of edge servers and the security of tasks in auction is considered. However, the two current methods for securing data, encryption and monitoring the external behavior of nodes, are not applicable to our system. Therefore, to identify untrustworthy edge servers in task auction and avoid using them to process tasks, we design a trust evaluation method based on the true utility acquisition. After being processed by edge servers, the results of tasks are put into applications and bring utility for devices. The true utility is related to the quality of task processing: if the edge server finishing processing before task deadline, the true utility meets the utility expected by devices before offloading. Otherwise, when the tasks are discarded maliciously or beyond deadline, the true utility is lower. However, due to the incapability of devices of communicating to the applications, the true utility should be sent to the corresponding UAV first and issued to the devices when the UAV passes them the next time. We assume that in the current round of task offloading the true utility of tasks in the last round has been known to the UAVs. Therefore, the UAVs can issue the true utility to devices when collecting the new tasks. And by comparing the true utility and expected utility, the task processing quality can be evaluated by UAV, and the trust value of edge servers can be obtained and updated.

$$\vartheta_d^{ij} = \frac{\left|\widehat{v}_d^{ij} - v_d^{ij}\right|}{\widehat{v}_d^{ij}}. \tag{35}$$

$$V_w' = \begin{cases} V_w + V_w \lambda_{IT} \vartheta_d^{ij}, \hat{v}_d^{ij} \geqslant \hat{v}_d^{ij}, \\ V_w - V_w \lambda_{DT} \vartheta_d^{ij}, \hat{v}_d^{ij} < \hat{v}_d^{ij}. \end{cases} \tag{36}$$

In Eq. (35), ϑ_d^{ij} denotes the difference between the expected utility of d_i for task t_j^i and the true utility after processing, where $\widehat{v}_d^{ij}$ is the expected utility, and v_d^{ij} is the true utility brought to d_i. In the case that the information provided by the application to the UAV is true, if the server s_w is trusted, s_w will finish t_j^i and return the result in time or even ahead of schedule, i.e. $v_d^{ij} \geq \widehat{v}_d^{ij}$. Conversely, if s_w maliciously discards t_j^i or the result is returned beyond the deadline, $v_d^{ij} < \widehat{v}_d^{ij}$. For both cases, the trust value update methods Increase Trust (IT) and Decrease Trust (DT) are proposed, as shown in Eq. (36). V_w and $V_w^{'}$ is the trust value before and after update, and λ_{IT} and λ_{DT} are the update strength of two update methods, respectively. Initially, all edge servers involved in task offloading are assigned an identical trust value V_{init}. Then UAVs use different update methods and strength according to the task processing performance of edge servers. The trust value gap between servers gradually becomes larger, and untrustworthy servers can be screened out, which guides the offloading decisions in subsequent process.

The trust above is a direct trust, i.e., the UAV responsible for d_i has direct interaction with s_w and can evaluate the trust value. However, the scope of direct trust evaluation for one UAV is limited, and identifying malicious servers only by direct trust is not efficient enough. For example, edge server s_1 is involved in task auction in work area a_1, so the UAV u_1 generates a direct trust value for s_1. If server s_1 is identified as untrustworthy, u_1 will not select it as offloading target during subsequent offloading. However, u_2, the UAV responsible for work area a_2, have no direct interaction with s_1 and therefore does not know that it is untrustworthy. In the next round, s_1 may join the auction hosted by u_2, affecting the task security. To address the situation and extend the trust evaluation scope for each UAV, we also introduce recommended trust [44] in the method. For each UAV, after the direct trust evaluation for any server, the evaluation result is recommended to other UAVs, and the UAVs can combine the direct trust evaluated by themselves and the recommended trust to derive the final trust value of servers. Eq. (37) illustrates the final trust value V_w^j of s_w evaluated by u_j, where V_w^j is the direct trust generated by u_j for s_w, V_w^k is the direct trust generated by other UAVs for s_w, and θ_d and θ_r are the weight of direct trust and recommended trust, respectively.

$$V_w^j = \theta_d V_w^j + \theta_r \frac{\sum_{u_k \in U, u_k \neq u_j} V_w^k}{|U| - 1}. \tag{37}$$

Algorithm 4 shows the main process of STMTO after integrating trust evaluation. Before task offloading, multi-UAVs perform working area division and task trajectory design based on the location of devices. UAVs collaborate to shorten the task processing delay. In the offloading stage, UAVs act as task offloaders, auctioneers and trust evaluators at the same time, ensuring the security of tasks and utility of both devices and edge servers.

Algorithm 4 Main Process of STMTO

```
Algorithm 1: Working area division for multi-UAVs
Algorithm 2: Trajectory plan for multi-UAVs
While r < R:
  For u_j in U:
    For d_i in a_j:
      Remove each s_w with trust value lower than V_thres
      from seller party
      Algorithm 3: Task auction for d_i
      u_j collects v_d^ik for each t_j^i and updates trust value of
      servers
      u_j recommends trust value
    End for
  End for
End While
```

5. Performance analysis

5.1. Experiment setup

In experiments, a network area of 1000m × 1000m with 100 devices randomly deployed is simulated. Multi-UAV system contains 5 homogeneous UAVs for offloading the tasks. In addition, there are 30 edge servers probably being involved in the task auction. The algorithms proposed above are implemented by Python, in which the behavior of data transmission, task auction and trust evaluation are simulated. Then we analyze the performance of the system in terms of offloading cost, task processing delay, task processing success rate and utility. During the experiments, the topology of the network is changed repeatedly to demonstrate the robustness of STMTO. Finally, we compare the performance of STMTO with other offloading strategies.

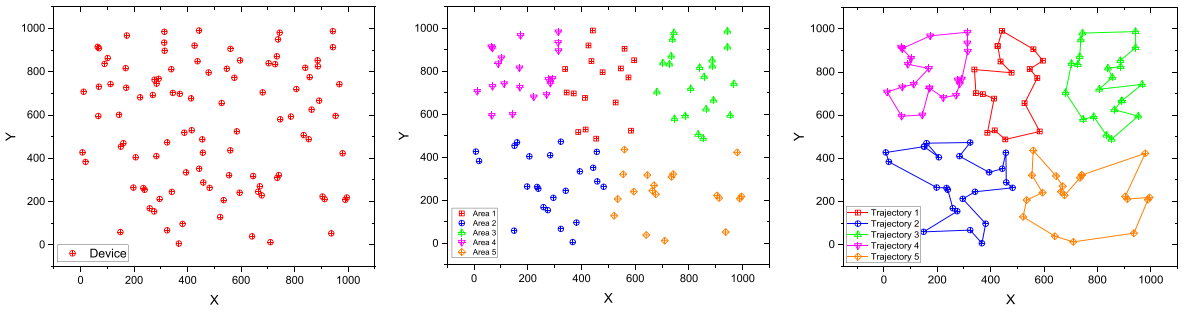
5.2. Performance of collaborative task collection scheme

Firstly, we analyze the performance of the collaborative task collection scheme for MUS. Fig. 3(a) shows the deployment of devices in the network area. Then the working area of UAVs is divided applying Algorithm 1. As shown in Fig. 3(b), the devices are divided into 5 subsets according to the physical location of devices. The devices in one working area are marked with same color, the tasks of which are offloaded by the same UAV. Fig. 3(c) illustrates trajectory design for MUS by Algorithm 2. It can be seen that the output trajectory of UAV is basically closed-loop, which achieves the goal of shortening the flight distance.

Fig. 4 illustrates the performance of trajectory design applying Algorithm 2. From (a), we see that the trajectory length of each UAV decreases as iteration times increases and finally converges. The output trajectory after 100 iterations is shown in Fig. 3(c). Fig. 4(b) shows the effect of the initial trajectory number, i.e., the capacity of G, on the trajectory optimization. From the figure, it is seen that as the initial trajectory number increases, the trajectory length decreases faster and the optimization effect is better. This is because a larger initial trajectory number increases the probability of generating better permutations and provides a larger solution space for trajectory optimization. However, too many trajectory numbers will reduce the efficiency of algorithm, so in the actual optimization process, we comprehensively consider the optimization results and the algorithm efficiency to choose the initial trajectory number. Fig. 4(c) shows the effect of different copy rate ψ. As ψ increases from 0, the optimization performance is more significant because some of ineffective trajectories in G are replaced by the best

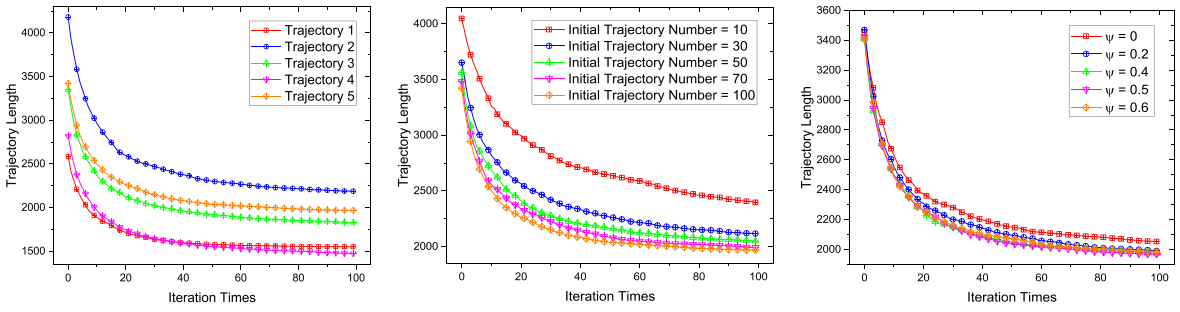
one. The overall quality of the trajectories in G improves, providing a better basis for the next iteration of optimization. However, the optimization performance becomes worse when ψ continues to increase, because too large ψ reduces the diversity of trajectories in G and makes the algorithm easily fall into local optimal solutions. From the analysis above, the parameter value in the practical trajectory design is shown in Table 1.

Fig. 5 shows the performance comparison of Single UAV (SU) and Multi-UAV System (MUS). From (a) we see that the trajectory optimization of SU has a long convergence time and it is difficult to obtain the optimal task collection trajectory due to the large number of devices. In contrast, in collaborative task collection strategy, the devices have been divided into different working areas for MUS, and each UAV can apply Algorithm 2 to design the collection trajectory for less devices simultaneously, which not only shortens the time of algorithm iteration, but also simplifies the trajectory of UAV. (b) compares the average waiting delay of tasks when using SU and MUS. SU takes a long time to complete a round of task offloading. Therefore, the tasks of back-ranked devices in its collection plan have longer waiting delay. However, multiple UAVs in MUS collaborate to share the total tasks in a balanced manner, which significantly reduces the time of one round of task offloading the average waiting delay of tasks is less. (c) shows the comparison of energy consumption applying SU and MUS, including



(a)The distribution of devices (b)The work area division for UAVs (c)The trajectory plan for UAVs

Fig. 3. The performance of working area division and trajectory plan for MUS.



(a)The trajectory length of different UAVs

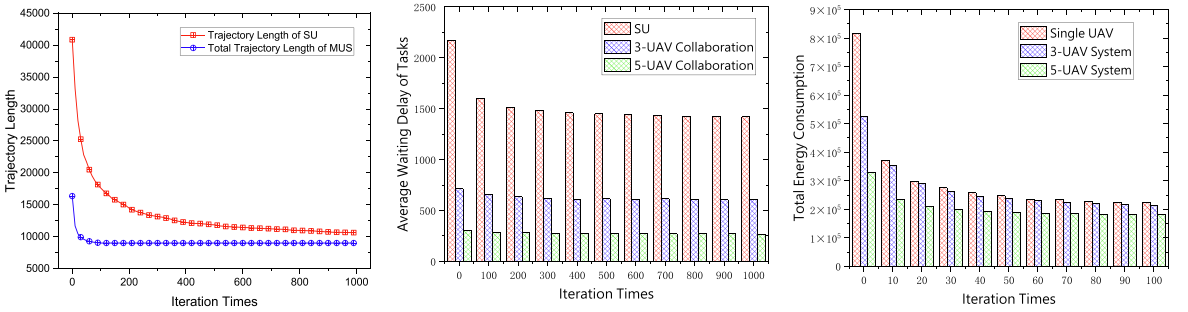
(b)The optimization performance with different initial trajectory number

(c)The optimization performance with different copy rate

Fig. 4. The effect of parameter value in Algorithm 2 on trajectory length.

Table 1
The parameter value in Algorithm 2.

Parameter	Value
Iteration times$R$	100
Initial trajectory number$\|G\|$	100
Copy rateψ	0.5



(a)The trajectory length of applying SU and MUS

(b) The average waiting delay of tasks applying SU and MUS

(c) The energy consumption of UAV applying SU and MUS

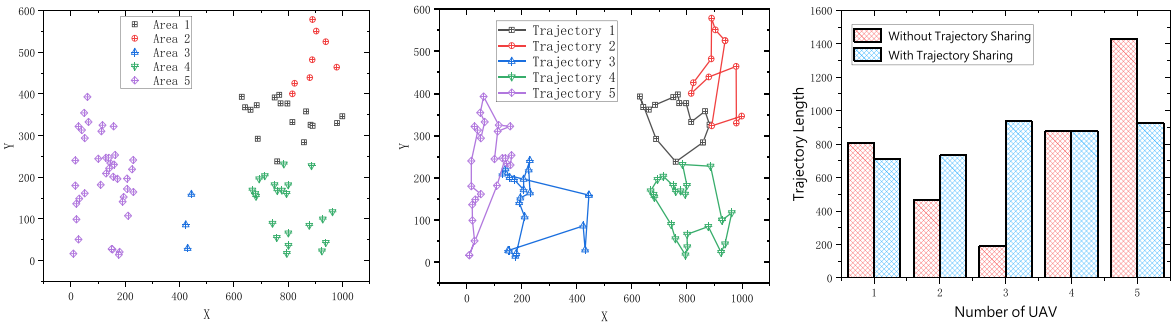
Fig. 5. The performance comparison of SU and MUS.

the energy of flight and data transmission. In the same scenario, the energy consumption difference between SU and MUS is mainly in the flight energy, and the variation of energy consumption with algorithm iterations corresponds to Fig. 5(a).

Fig. 6 illustrates the performance of dynamic trajectory sharing strategy for the special situation where devices are deployed unevenly. (a) shows the work area division in such scenario directly applied Algorithm 1. We can see the division is not uniform by comparing the device number in Area 3 and Area 5. After using the dynamic trajectory sharing strategy, the actual collection trajectories are shown in (b). From (b), u_3 is responsible for a small number of devices in the working area a_3, while in a_5 the devices are densely distributed. In the actual collection process u_3 can dynamically change the route according to the completion of u_5 and traverse the remaining devices in a_5 in reverse. (c) compares the trajectory length between the modified strategy and original one. From the figure, it is clear that original strategy does not work well for the special case of uneven distribution of devices, but after the modification, the flight length of UAVs is significantly uniform.

5.3. Performance of trust based task auction mechanism

Fig. 7 illustrates the performance of the double auction mechanism. (a) shows the effect of the parameter k_d^{ij} and k_w^{ij} in the public bid calculation on the utility of device d_i and edge server s_w when only one task needs to be auctioned. The solid black curve in (a) indicates the function image of the desired utility $E\left(U_d^{ij}\right)$, and the red curve indicates the real utility of d_i in the

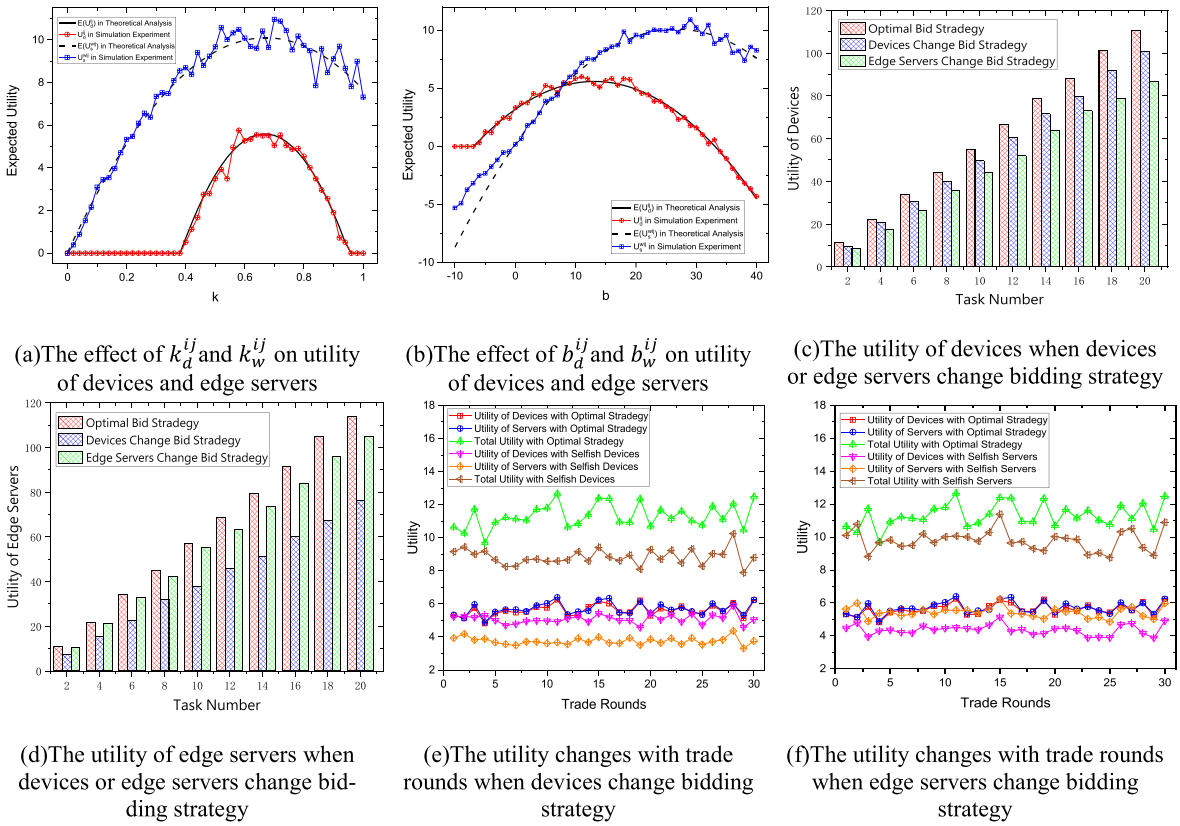


(a)The work area division in the network with uneven distribution of devices

(b)The trajectories of UAVs with dynamic trajectory sharing

(c)The trajectory length of each UAV without and with dynamic trajectory sharing

Fig. 6. The performance of dynamic trajectory sharing strategy for the special situation.



(a)The effect of k_d^{ij} and k_w^{ij} on utility of devices and edge servers

(b)The effect of b_d^{ij} and b_w^{ij} on utility of devices and edge servers

(c)The utility of devices when devices or edge servers change bidding strategy

(d)The utility of edge servers when devices or edge servers change bidding strategy

(e)The utility changes with trade rounds when devices change bidding strategy

(f)The utility changes with trade rounds when edge servers change bidding strategy

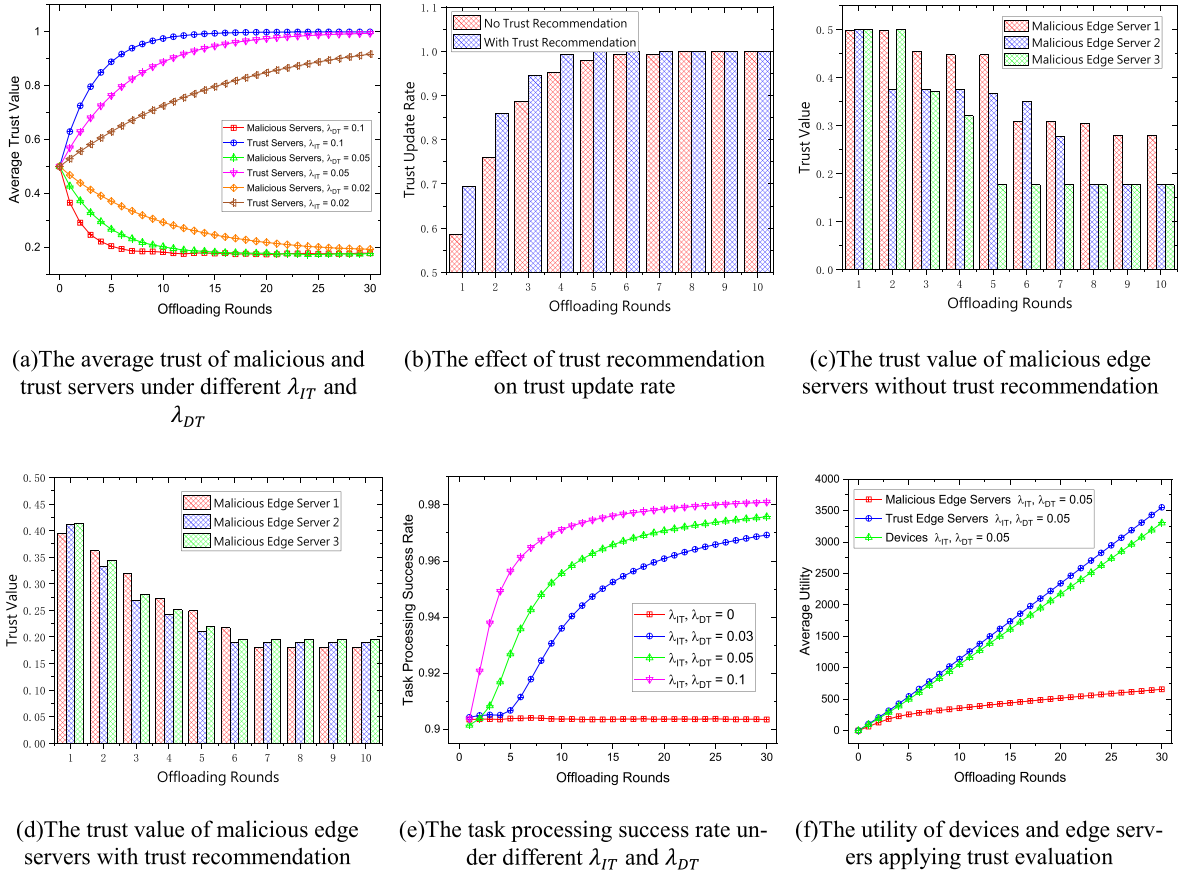
Fig. 7. The performance of auction scheme based on optimal bidding strategy.

simulated experiment. The trend of the two curves basically coincides. When k_d^{ij} is too small, the public bid of d_i is too low, resulting in the trade principle $z^{ij} < v_d^{ij}$ in Algorithm 3 not being met, so the trade fails and the utility of d_i is 0. As k_d^{ij} increases, the utility keeps increasing. However, when k_d^{ij} is too large it leads to a smaller gap between the private valuation and the public bid of d_i, and its utility gradually decreases. Therefore, the utility of d_i reaches the maximum when $k_d^{ij} = k_d^{ij^*}$. Similarly, we can analyze that the utility of s_w is maximized when $k_w^{ij} = k_w^{ij^*}$. (b) shows the impact of parameter b_d^{ij} and b_w^{ij} in the public bid calculation on the utility. Similar to (a), the function of expected utility with b_d^{ij} and b_w^{ij} is also a convex function, and the

optimal parameter value $b_d^{ij^*}$ and $b_w^{ij^*}$ is obtained at the horizontal coordinate corresponding to the highest point of the curves for d_i and s_w respectively.

Fig. 7(c) and (d) show the impact of deviation from the optimal bidding strategy on the utility of devices and edge servers when there are multiple tasks and servers in the auction environment. As seen in (c), the utility of devices increases with task number. In addition, the devices have the maximum utility when both parties follow the optimal bidding strategy, while the utility of devices decreases when devices or edge servers selfishly deviate from the optimal bid. Similarly, (d) also illustrates selfish bid is not beneficial to the utility of edge servers. Then the impact of selfish bid on the utility balance is analyzed. In (e) we see that when devices and edge servers follow the optimal bidding strategy, the utility of both is maximized and there is no significant difference between them, as shown by the red and blue curves, indicating that the optimal bidding strategy is fair and the utility balance is reached. However, when devices deviate from the optimal bid, the utility of both decreases, with utility of edge servers being more severely impaired and the total utility of system decreasing accordingly. Similarly, (f) shows that the selfish bid of edge servers also reduces the system utility. From the experimental analysis, the optimal bidding strategy makes the utility of both devices and edge servers maximum and balanced. Instead, the two parties only focusing on their own utility and selfishly bidding do not enhance either their own utility or utility of opponent. In summary, the UAVs controlling the range of public bid and the optimal bidding strategy eliminate malicious bids in auction process and improve the matching degree of bids between devices and edge servers, thus the utility of both parties in task offloading can reach optimization at the same time.

Next we analyze the performance of auction mechanism integrating trust evaluation in the case of partial untrustworthy edge servers, as shown in Fig. 8. From (a), we can see the variation of average trust value of the trusted and malicious edge servers under different update strength λ_{IT} and λ_{DT}. As the offloading rounds increase, the trust value of trusted servers improves because they process the tasks with results matching the expected utility of devices. On the contrary, the trust value of malicious servers gradually decreases. Moreover, in the evaluation, the greater the update strength, the greater the change in trust value. According to the trust value calculation formula, the trust value should take a range between 0 and 1. However, the UAVs need to evaluate the trust value of servers when selecting the server to offload tasks. When the trust value of servers is lower than 0.2, i.e., $V_{thres} = 0.2$ in Algorithm 4, the servers will not be selected for task processing.



(a)The average trust of malicious and trust servers under different λ_{IT} and λ_{DT}

(b)The effect of trust recommendation on trust update rate

(c)The trust value of malicious edge servers without trust recommendation

(d)The trust value of malicious edge servers with trust recommendation

(e)The task processing success rate under different λ_{IT} and λ_{DT}

(f)The utility of devices and edge servers applying trust evaluation

Fig. 8. The performance of trust evaluation mechanism.

Once there is no interaction between the servers and devices, its trust value will not be updated in subsequent rounds either. Therefore, the lower bound of the trust value of malicious servers in (a) is around 0.2. The update of trust value in (a) is combined direct trust and recommended trust, and then we analyze the impact of trust recommendation of MUS on the trust evaluation performance. (b) shows the difference in the trust update rate with and without trust recommendation. The trust update rate reflects the scope of trust evaluation, and it is maximized when the trust value of each server has been updated for each UAV. We see that the trust update rate grows more slowly when only direct trust is available compared to adding recommended trust. This is because direct trust can only be obtained from the direct interaction of UAV with servers. As can also be seen from the example in Section 4.3, the malicious server may only participate in the auction hosted by some UAVs, so the other UAVs do not have direct interaction with the malicious server and have no direct trust for it. When the malicious server selects the auction of the UAV that did not interact with it before, the UAV cannot immediately identify the malicious server only by direct trust. (c) and (d) show the trust update of some UAV for three malicious servers only by direct trust and recommended trust combined, respectively. Comparing (c) with (d), we find that when there is no trust recommendation, the trust value of malicious servers is updated slowly, and the trust evaluation adding recommended trust identifies the malicious servers much faster. Thus, trust recommendation not only expands the scope of trust update but also improves the sensitivity of identifying malicious nodes. Moreover, the effect of trust evaluation on task processing and utility is analyzed. (e) shows the variation of task processing success rate R with the update strength λ_{IT} and λ_{DT}. We see that when there is no trust evaluation, i.e., $\lambda_{IT} = 0$, $\lambda_{DT} = 0$, R is always low and does not change significantly. As λ_{IT} and λ_{DT} gradually increases, R raises with offloading rounds, indicating that with the trust evaluation malicious servers are identified, and more tasks are offloaded to trusted servers, resulting in more tasks that achieve the expected utility of devices. Corresponding to (a), the larger the value of λ_{IT} and λ_{DT}, the more sensitive the trust evaluation is and the more significant R boost. (f) reveals the change in the utility of devices, trusted servers and malicious server when trust evaluation is added. As the offloading rounds increases, the malicious servers have less chance to be selected as offloading targets. Therefore, the utility increment per round gradually decreases. On the contrary, the utility of trusted servers and devices remain on the same increasing trend and balanced. The above analysis demonstrates the effectiveness of the task auction mechanism integrating trust evaluation in ensuring the utility and identifying malicious participants.

5.4. Comprehensive performance analysis of STMTO

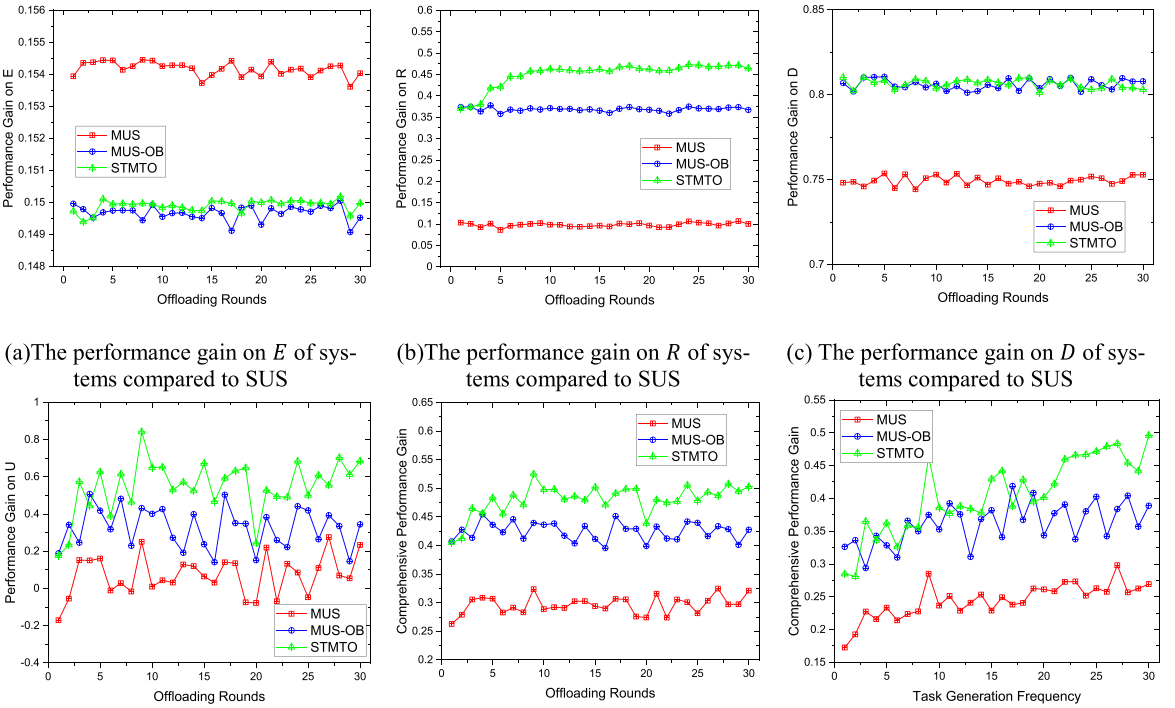
The above experimental analysis illustrates the performance of collaborative task collection strategy and trust based task auction mechanism on energy consumption, task processing delay and utility. In this subsection, we provide a comprehensive analysis for the performance of STMTO. Since STMTO is a novel task offloading architecture, there is no other corresponding systems to directly compare with it. For the improving aspects focused by STMTO, such as bidding strategy and trust evaluation, we extract features from the strategies proposed in [25] and [35], to construct the benchmark system and other comparison systems, which are listed in Table 2. We take the performance of SUS as the benchmark and calculate the performance gain of STMTO and other offloading systems based on the optimization goals in Problem Statement, including total system energy consumption E, total task processing delay D, task processing success rate R, and total utility U. The calculating method of performance gain ΔP is shown in Equation (38). The performance gain of the schemes is shown in Fig. 9.

$$\Delta P = \begin{cases} \frac{v_P^b - v_P^o}{v_P^b}, P = EorD, \\ \frac{v_P^o - v_P^b}{v_P^b}, P = RorU. \end{cases} \tag{38}$$

As seen in (a), compared to the benchmark system, the other three systems including STMTO have a reduction in total energy consumption. In addition, the performance gain of MUS is slightly larger than that of MUS-OB and STMTO, which is due to the fact that MUS does not adopt the optimal bidding strategy, and the number of failed tasks due to selfish bids from devices and edge servers is increased. Therefore, the energy consumption of processing tasks is reduced. (b) shows the comparison of the four systems in terms of task processing success rate. Among them, failed tasks include tasks that exceed the deadline, tasks in failed auction, and tasks whose true utility are lower than the expected utility. Multi-UAV collaborative task collection solves the problem that the long collecting time of SUS causes many tasks to reach deadline and expire. Therefore, the performance gain of all three systems is greater than 0. The task processing success rate of MUS is smaller than that of MUS-OB and STMTO, indicating that the selfish bids bring more failure of auction, which also corresponds to the energy

Table 2
Different schemes and their features.

System	Number of UAVs	Bidding strategy	Trust evaluation
SUS [25]	1	Selfish	No
MUS [35]	5	Selfish	No
MUS-OB [35]	5	Optimal	No
STMTO	5	Optimal	Yes



(a)The performance gain on E of systems compared to SUS

(b)The performance gain on R of systems compared to SUS

(c) The performance gain on D of systems compared to SUS

(d)The performance gain on U of systems compared to SUS

(e)The comprehensive performance gain of systems compared to SUS

(f)The impact of task generation frequency on comprehensive performance gain

Fig. 9. The performance comparison of different task offloading systems.

consumption in (a). Compared with MUS-OB, the task processing success rate of STMTO keeps increasing with offloading rounds, which is because the trust evaluation mechanism has gradually identified malicious servers and the number of discarded tasks gradually decreases. (c) shows the performance gain on task processing delay, from which the multi-UAV cooperative offloading significantly reduces the task processing delay compared to SUS. Moreover, we see that the lower performance gain of MUS is due to the fact that selfish bids lead to failed trades and tasks linger longer to be processed. (d) shows the comparison of the device utility under different systems. MUS-OB using optimal bidding strategy significantly outperforms selfish bids, while in STMTO trust evaluation of edge servers reduces the probability of malicious servers processing tasks, further improves the quality of task processing and ensures the utility of devices. (e) and (f) analyze the comprehensive performance gain of systems, it can be seen that STMTO has the best comprehensive performance, and the performance gain increases with the task generation frequency, thus STMTO is more suitable for scenarios where tasks occur frequently.

6. Conclusion and future work

Task offloading, as an emerging computing mode, solves the problem that the task complexity is beyond the computing capacity and energy of IoT devices by using the residual resource of edge servers. Moreover, UAV-enhanced task offloading system has been widely studied in scenarios where devices have no direct access to the Internet. The STMTO proposed in this paper applies multi-UAV system (MUS) to task offloading. Based on the idea of K-means algorithm and genetic algorithm, a collaborative task collection strategy for multiple UAVs is designed, which balances the energy consumption of the system and reduces the waiting delay of tasks. Secondly, we consider the utility relationship between devices and edge servers and task security issues, and establish a trust based double task auction mechanism. The UAVs act as auctioneers to make offloading decisions based on the public bids and trust value of edge servers. According to our optimization goals a static game model with incomplete information is established and the optimal bidding strategy for both parties is obtained through theoretical analysis, which maximizes the utility of devices and edge servers. In addition, to ensure the task security and processing efficiency, the collaborative trust evaluation strategy is integrated in task auction to screen trusted edge servers as offloading targets, avoiding tasks are dropped by malicious servers. In experimental analysis, compared to the benchmark system, our proposed STMTO has a comprehensive performance improvement of 48.02% in four aspects: energy consumption, task processing delay, task processing success rate and offloading utility, which proves the effectiveness of

our system. In the future, we will take the task generation frequency of heterogeneous devices into consideration to improve the collaborative collection strategy for MUS, allowing UAVs to dynamically optimize trajectories during task collection based on the location and task number of uncovered devices in the own working area and other working areas.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (No. 62072475, No. 61772554), the Fundamental Research Funds for the Central Universities, JLU, the Natural Science Foundation of Hunan Province under Grant 2020JJ4237.

References

[1] A.S. Deese, J. Jesson, T. Brennan, S. Hollain, P. Stefanacci, E. Driscoll, C. Dick, K. Garcia, R. Mosher, B. Rentsch, A. Bechtel, E. Rodriguez, Long-Term Monitoring of Smart City Assets via Internet of Things and Low-Power Wide-Area Networks, IEEE Internet Things J. 8 (1) (2021) 222–231.
[2] M. Cicioglu, A. Calhan, Internet of Things-Based Firefighters for Disaster Case Management, IEEE Sens. J. 21 (1) (2021) 612–619.
[3] F. Li, G. Huang, Q. Yang, M. Xie, Adaptive Contention Window MAC Protocol in a Global View for Emerging Trends Networks, IEEE Access 9 (2021) 18402–18423.
[4] K.L. Lueth, State of the IoT 2018: Number of IoT devices now at 7B-Market accelerating, IoT Analytics (2018).
[5] M. Yu, A. Liu, N. Xiong, T. Wang, An Intelligent Game based Offloading Scheme for Maximizing Benefits of IoT-Edge-Cloud Ecosystems, IEEE Internet Things J. (2020), https://doi.org/10.1109/JIOT.2020.3039828.
[6] W. Huang, K. Ota, M. Dong, T. Wang, S. Zhang, J. Zhang, Result Return Aware Offloading Scheme in Vehicular Edge Networks for 6G driving Application, Comput. Commun. 164 (2020) 201–214.
[7] X. Zhu, Y. Luo, A. Liu, M.Z.A. Bhuiyan, S. Zhang, Multi-Agent Deep Reinforcement Learning for Vehicular Computation Offloading in IoT, IEEE Internet Things J. (2020), https://doi.org/10.1109/JIOT.2020.3040768.
[8] H. Qiu, M. Qiu, M. Liu, G. Memmi, Secure health data sharing for medical cyber-physical systems for the healthcare 4.0, IEEE journal of biomedical and health informatics 24 (9) (2020) 2499–2505.
[9] M. Shen, A. Liu, G. Huang, N.N. Xiong, H. Lu, ATTDC: An Active and Trace-able Trust Data Collection Scheme for Industrial Security in Smart Cities, IEEE Internet Things J. 8 (8) (2021) 6437–6453, https://doi.org/10.1109/JIoT.648890710.1109/JIOT.2021.3049173.
[10] S. Huang, Z. Zeng, K. Ota, M. Dong, T. Wang, N.N. Xiong, An Intelligent Collaboration Trust Interconnections System for Mobile Information Control in Ubiquitous 5G networks, IEEE Trans. Network Sci. Eng. 8 (1) (2021) 347–365.
[11] A. Li, W. Liu, L. Zeng, C. Fa, Y. Tan, An Efficient Data Aggregation Scheme based on Differentiated Threshold Configuring Joint Optimal Relay Selection in WSNs, IEEE Access 9 (2021) 19254–19269.
[12] A. Liu, Z. Chen, N. Xiong, An Adaptive Virtual Relaying Set Scheme for Loss-and-Delay Sensitive Wireless Sensor Networks, Inf. Sci. 424 (2018) 118–136.
[13] Y. Ouyang, A. Liu, N. Xiong, T. Wang, An Effective Early Message Ahead Join Adaptive Data Aggregation Scheme for Sustainable IoT, IEEE Trans. Network Sci. Eng. 8 (1) (2021) 201–219.
[14] Z. Tian, M. Li, M. Qiu, Y. Sun, S. Su, Block-DEF: A secure digital evidence framework using blockchain, Inf. Sci. 491 (2019) 151–165.
[15] X. Liu, H. Song, A. Liu, Intelligent UAVs Trajectory Optimization from Space-Time for Data Collection in Social Networks, IEEE Trans. Network Sci. Eng. (2020), https://doi.org/10.1109/TNSE.2020.3017556.
[16] X. Zhu, Y. Luo, A. Liu, W. Tang, M.Z.A. Bhuiyan, A Deep Learning-Based Mobile Crowdsensing Scheme by Predicting Vehicle Mobility, IEEE Trans. Intell. Transp. Syst. (2020), https://doi.org/10.1109/TITS.2020.3023446.
[17] T. Li, A. Liu, S. Zhang, T. Wang, N. Xiong, A Trustworthiness-based Vehicular Recruitment Scheme for Information Collections in Distributed Networked Systems, Inf. Sci. 545 (2021) 65–81.
[18] D. Pliatsios, P. Sarigiannidis, T. Lagkas, et al, A survey on SCADA systems: secure protocols, incidents, threats and tactics, IEEE Commun. Surv. Tutorials 22 (3) (2020) 1942–1976.
[19] M.M. Hasan, H.T. Mouftah, Optimization of trust node assignment for securing routes in smart grid SCADA networks, IEEE Syst. J. 13 (2) (2019) 1505–1513.
[20] J. Guo, A. Liu, K. Ota, M. Dong, X. Deng, N. Xiong, ITCN, An Intelligent Trust Collaboration Network System in Industrial IoT,", IEEE Trans. Network Sci. Eng. (2021), https://doi.org/10.1109/TNSE.2021.3057881.
[21] Y. Liu, A. Liu, X. Liu, X. Huang, A Statistical Approach to Participant Selection in Location-based Social Networks for Offline Event Marketing, Inf. Sci. 480 (2019) 90–108.
[22] J. Guo, F. Li, T. Wang, S. Zhang, Y. Zhao, Parameters Analysis and Optimization of Polling-Based MAC Protocol for Multi-Sensors Communication, Int. J. Distrib. Sens. Netw. (2021), https://doi.org/10.1177/15501477211007412/.
[23] Y. Liu, T. Wang, S. Zhang, X. Liu, X. Liu, Artificial Intelligence Aware and Security-enhanced Trace-back Technique in Mobile Edge Computing, Comput. Commun. 161 (2020) 375–386.
[24] Wenjuan Tang, Ju Ren, Kun Deng, Yaoxue Zhang, Secure data aggregation of lightweight e-healthcare iot devices with fair incentives, IEEE Internet Things J. 6 (5) (2019) 8714–8726.
[25] H. Qiu, Q. Zheng, G. Memmi, J. Lu, M. Qiu, B. Thuraisingham, Deep Residual Learning-Based Enhanced JPEG Compression in the Internet of Things, IEEE Trans. Ind. Inf. 17 (3) (2020) 2124–2133.
[26] Taewoon Kim, Daji Qiao, Energy-Efficient Data Collection for IoT Networks via Cooperative Multi-Hop UAV Networks, IEEE Trans. Veh. Technol. 69 (11) (2020) 13796–13811.
[27] Changqin Huang, Guosheng Huang, Wei Liu, Ruoyu Wang, Mande Xie, A parallel joint optimized relay selection protocol for wake-up radio enabled WSNs, Phys. Commun. 47 (2021) 101320, https://doi.org/10.1016/j.phycom.2021.101320.
[28] T. Li, W. Liu, Z. Zeng, N.N. Xiong, DRLR: A Deep Reinforcement Learning based Recruitment Scheme for Massive Data Collections in 6G-based IoT networks, IEEE Internet Things J. (2021), https://doi.org/10.1109/JIOT.2021.3067904.
[29] H. Teng, M. Dong, Y. Liu, W. Tian, X. Liu, A low-cost physical location discovery scheme for large-scale Internet of Things in smart city through joint use of vehicles and UAVs, Future Generation Computer Systems 118 (2021) 310–326.
[30] Hongzhi Guo, Jiajia Liu, UAV-enhanced intelligent offloading for internet of things at the edge, IEEE Trans. Ind. Inf. 16 (4) (2020) 2737–2746.
[31] B. Jiang, G. Huang, T. Wang, J. Gui, X. Zhu, Trust based energy efficient data collection with unmanned aerial vehicle in edge network, Transactions on Emerging Telecommunications Technologies (2020), https://doi.org/10.1002/ett.3942 e3942.

[32] W. Chen, Z. Su, Q. Xu, T. H. Luan, R. Li. VFC-Based Cooperative UAV Computation Task Offloading for Post-disaster Rescue. IEEE INFOCOM 2020 - IEEE Conference on Computer Communications, Toronto, ON, Canada, 2020, pp. 228-236, doi: 10.1109/INFOCOM41043.2020.9155397.
[33] Y. Ouyang, Z. Zeng, X. Li, T. Wang, X. Liu, A Verifiable Trust Evaluation Mechanism for Ultra-Reliable Applications in 5G and Beyond Networks, Computer Standards & Interfaces 77 (2021) 103519.
[34] Arun Das, Shahrzad Shirazipourazad, David Hay, Arunabha Sen, Tracking of multiple targets using optimal number of UAVs, IEEE Trans. Aerosp. Electron. Syst. 55 (4) (2019) 1769–1784.
[35] Pengfei Wu, Fu Xiao, Haiping Huang, Ruchuan Wang, Load balance and trajectory design in multi-UAV aided large-scale wireless rechargeable networks, IEEE Trans. Veh. Technol. 69 (11) (2020) 13756–13767.
[36] Qiyu Hu, Yunlong Cai, Guanding Yu, Zhijin Qin, Minjian Zhao, Geoffrey Ye Li, Joint Offloading and Trajectory Design for UAV-Enabled Mobile Edge Computing Systems, IEEE Internet Things J. 6 (2) (2019) 1879–1892, https://doi.org/10.1109/JIoT.648890710.1109/JIOT.2018.2878876.
[37] J. Xiong, H. Guo, J. Liu, Task Offloading in UAV-Aided Edge Computing: Bit Allocation and Trajectory Optimization, IEEE Commun. Lett. 23 (3) (2019) 538–541, https://doi.org/10.1109/LCOMM.2019.2891662.
[38] L. Zhang, Z. Zhao, Q. Wu, H. Zhao, H. Xu, X. Wu, Energy-Aware Dynamic Resource Allocation in UAV Assisted Mobile Edge Computing Over Social Internet of Vehicles, IEEE Access 6 (2018) 56700–56715, https://doi.org/10.1109/ACCESS.2018.2872753.
[39] Sudip Misra, Bernd E. Wolfinger, M.P. Achuthananda, Tuhin Chakraborty, Sankar N. Das, Snigdha Das, Auction-Based Optimal Task Offloading in Mobile Cloud Computing, IEEE Syst. J. 13 (3) (2019) 2978–2985, https://doi.org/10.1109/JSYST.426700310.1109/JSYST.2019.2898903.
[40] Farshad Mashhadi, Sergio A. Salinas Monroy, Arash Bozorgchenani, Daniele Tarchi, Optimal auction for delay and energy constrained task offloading in mobile edge computing, Comput. Netw. 183 (2020) 107527, https://doi.org/10.1016/j.comnet.2020.107527.
[41] Yong Wang, Zhi-Yang Ru, Kezhi Wang, Pei-Qiu Huang, Joint Deployment and Task Scheduling Optimization for Large-Scale Mobile Users in Multi-UAV-Enabled Mobile Edge Computing, IEEE Trans. Cybern. 50 (9) (2020) 3984–3997, https://doi.org/10.1109/TCYB.622103610.1109/TCYB.2019.2935466.
[42] Lei Yang, Haipeng Yao, Jingjing Wang, Chunxiao Jiang, Abderrahim Benslimane, Yunjie Liu, Multi-UAV-Enabled Load-Balance Mobile-Edge Computing for IoT Networks, IEEE Internet Things J. 7 (8) (2020) 6898–6908, https://doi.org/10.1109/JIoT.648890710.1109/JIOT.2020.2971645.
[43] Wenjuan Tang, Ju Ren, Yaoxue Zhang, Enabling trusted and privacy-preserving healthcare services in social media health networks, IEEE Trans. Multimedia 21 (3) (2019) 579–590.
[44] M. Al-khafajiy, T. Baker, M. Asim, Z. Guo, R. Ranjan, A. Longo, D. Puthal, M. Taylor, COMITMENT: A Fog Computing Trust Management Approach, J. Parallel Distrib. Comput. 137 (2020) 1–16, https://doi.org/10.1016/j.jpdc.2019.10.006.
[45] Wenjuan Tang, Ju Ren, Kuan Zhang, Deyu Zhang, Yaoxue Zhang, Xuemin (Sherman) Shen, Efficient and privacy-preserving fog-assisted health data sharing scheme, ACM Transactions on Intelligent Systems and Technology (TIST) 10 (6) (2019) 1–23.
[46] N. Xiong, A.V. Vasilakos, L.T. Yang, L. Song, Y. Pan, Y. Li, Comparative Analysis of QoS and Memory Usage of Adaptive Failure Detectors in Healthcare Systems, IEEE J. Sel. Areas Commun. 27 (4) (2009) 495–509.
[47] K. Cheng, Y. Teng, W. Sun, A. Liu, X. Wang. Energy-Efficient Joint Offloading and Wireless Resource Allocation Strategy in Multi-MEC Server Systems. 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, 2018, pp. 1-6, doi: 10.1109/ICC.2018.8422877.