Stackelberg-Game-Based Intelligent Offloading Incentive Mechanism for a Multi-UAV-Assisted Mobile-Edge Computing System

Maoli Wang, Lu Zhang, Peng Gao, Xiaoli Yang, Kang Wang, and Kunlun Yang

***Abstract*—We study the intelligent offloading problem for a multiple unmanned aerial vehicle (multi-UAV)-assisted mobile-edge computing (MEC) system in an MEC scenario where a natural disaster has damaged the edge server. The study has two steps. First, the task offloading destination is determined by minimizing the total energy consumption of the multi-UAVs in the system. We propose the server selection game-theoretic (SSGT) algorithm and demonstrate its convergence through simulation experiments. Second, we propose an offloading incentive mechanism to price computing resources for a single unmanned aerial vehicle (UAV)-MEC server. Considering the UAV's power consumption and mobile users' willingness, we model the interaction between the UAV-MEC server and mobile users as a Stackelberg game. We prove the existence of a Nash equilibrium by theoretical analysis and experimental verification and design the multiround iterative game (MRIG) algorithm based on arithmetic descent to achieve the optimal solution, i.e., the utility tradeoff between the UAV-MEC server and mobile users. Finally, the simulation results show that our proposed scheme can increase the value of overall user satisfaction (SoU) more than other schemes, which proves that the incentive mechanism of resource pricing can supply computing power support for ground mobile users in a UAV-assisted MEC system more effectively.**

***Index Terms*—Data offloading incentive mechanism, multiple unmanned aerial vehicle (multi-UAV)-assisted mobile-edge computing (MEC) system, Stackelberg game, task offloading destination.**

I. Introduction

With the rapid development of the Internet of Things (IoT) and the popularity of the 5th-generation mobile communication technology (5G) network, the era of the Internet of Everything has arrived [1]. Under the high transmission pressure of mobile cloud computing (MCC) [2], [3] in transmitting massive data, mobile-edge computing (MEC) [4] was developed. In MEC, the computational tasks of mobile users (referred to in this article as users) are offloaded to the edge server, and the transmission distance to the physical location is decreased. By providing real-time data analysis and intelligent processing nearby, the user's experience quality can be improved [5].

Manuscript received 29 September 2022; revised 27 January 2023; accepted 25 March 2023. Date of publication 13 April 2023; date of current version 24 August 2023. This work was supported in part by the Shandong Provincial Natural Science Foundation of China under Grant ZR202111260301, and in part by the Shandong Province Agricultural Major Application Technology Innovation Project of China under Grant SD2019NJ007. *(Corresponding author: Maoli Wang.)*

The authors are with the School of Cyber Science and Engineering, Qufu Normal University, Qufu 273165, China (e-mail: wangml@qfnu.edu.cn; zhangl9803@163.com; siriushawk@foxmail.com; 1819032051@qq.com; 527061219@qq.com; yangkunlunqfnu@163.com).

Digital Object Identifier 10.1109/JIOT.2023.3265432

However, the existing MEC technology cannot efficiently operate with the limited available infrastructure. When natural disasters occur and edge servers are damaged or partially damaged, the MEC system will not be sufficient to support the computation offloading requirements of users in its signal coverage area. In this case, unmanned aerial vehicles (UAVs) have drawn the attention of researchers due to their flexibility and easy deployment [6], [7], [8]. The UAV-assisted MEC model is designed and constructed by introducing UAVs with communication modules and computing modules into the MEC system [9], [10]. Cheng et al. [11] discussed the benefits and feasibility of UAV-assisted MEC system in edge caching and assisted computing.

The new setting of the MEC system for UAVs brings new challenges. Due to the limited physical size of a UAV [6], [12], the computing and storage resources of the UAV-MEC server are limited. The traditional MEC scenario also often faces insufficient resources. Compared with that in the traditional MEC scenario, the problem of insufficient resources in the UAV-assisted MEC scenario is more prominent and cannot be ignored. Faced with a large number of computing demands from multiple users, we can find an intelligent offloading scheme to improve the utilization of resources and alleviate the resource shortage to a certain extent.

Some prior related work was performed in [13], [14], [15], [16], [17], [20], [21], [22], and [23]. For example, Liu et al. [13] minimized user energy consumption by solving a mixed-integer nonconvex problem with resource and delay requirement constraints. Seid et al. [14] studied a cooperative computation offloading and resource allocation scheme for air-to-ground (A2G) network emergencies based on model-free deep reinforcement learning (DRL). Ei et al. [15] proposed an optimization scheme for joint task offloading, communication, and computing resource allocation. In [16], a new workflow model based on time division multiple access (TDMA) was designed due to the energy efficiency of the UAV. In addition, some scholars have researched issues, such as task delay [17], [18], [19], computational efficiency [20], UAV flight characteristics [21], offloading security and privacy [22], and task offloading of object-oriented applications [23] in the UAV-assisted MEC systems.

In previous studies, objective factors, such as energy consumption, latency, and privacy were mostly used as the measurement criteria, and few people have considered the subjective will of users. However, in the face of fierce competition for resources, the subjective willingness of users becomes particularly important. In this article, we focus on the subjective willingness of users and study computation offloading in a UAV-assisted MEC system, and we consider the offloading willingness of the UAV-MEC server. In the case of insufficient resources, we adopt an incentive mechanism based on resource pricing to transform the computation offloading process into an offloading incentive problem [24], [25]. First, according to actual UAV battery power and UAV-MEC server computing capability, we propose that the UAV-MEC server should be selfish and rational. The UAV-MEC server will not actively participate in offloading unless the remuneration is sufficient. If users care about improving user experience, they must pay the corresponding cost. In addition, we determine the offloading destination via preoffloading to complete the initial resource allocation process and alleviate the resource pressure during the actual offloading process. Thus, we need to answer the following three questions to achieve the optimal allocation of computing resources.

1) Which UAV does each user choose to offload data to?
2) How much data does each user intend to offload to the specified UAV-MEC servers?
3) How does the UAV-MEC server price computing resources for each user separately?

We choose game theory [26] to analyze the competition among interests and determine the tradeoff between users and the UAV-MEC server. Game theory has been widely used to study the interaction between decision makers in conventional MEC systems [27], [28], [29], [30], [31]. Mitsis et al. [27] and Fan et al. [28] transformed the computation offloading problem into a noncooperative game by considering the risk awareness of users and the computing delays of the tasks, respectively. Fang et al. [29] modeled the total offloading benefit as a game process by introducing game theory and proposed a return-based distributed multiuser task offloading algorithm. In [30], the computation offloading problem for the Internet of Vehicles was formulated as a distributed offloading decision-making game. In [31], an offloading algorithm based on mean-field games was proposed for superdense networks. Chen et al. [32] designed a game learning algorithm combining a double deep Q network (DDQN) and distributed long short-term memory (LSTM).

In this article, we first determine the task offloading destination of each user through game theoretic ideas. Then, we let each UAV-MEC server play a Stackelberg game individually with the users it covers in the system. We design an iterative algorithm to determine the optimal data offloading strategy and pricing strategy that satisfies both the UAV-MEC server and users. In summary, the main contributions of this article are as follows.

1) We first establish a multiple UAV (multi-UAV)-assisted MEC system and perform a task offloading destination simulation by minimizing the total energy consumption of all UAVs in the system. To solve this problem, we design the server selection game-theoretic (SSGT) algorithm and simulate the task offloading process through simulation experiments.

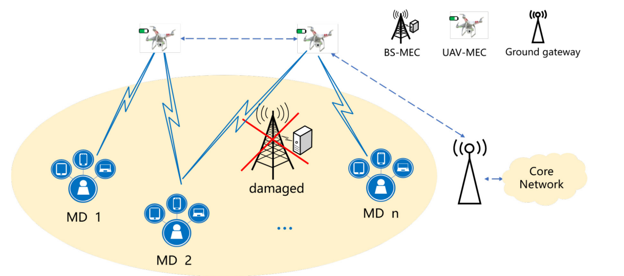

Fig. 1. Network architecture. The circle indicates the original BS signal coverage area.

2) After determining the task offloading destination, we introduce a computation offloading incentive mechanism of computing resource pricing to encourage the UAV-MEC server to actively participate in computation offloading. At the same time, to consider the subjective evaluations of users, we define the user satisfaction degree (USD) function to directly measure the offloading willingness of users.
3) We describe the interaction between the individual UAV-MEC server and users who plan to offload to it as a Stackelberg game to obtain the optimal strategy for all participants. We prove the existence and uniqueness of the Nash equilibrium through both theoretical and simulation experiments.
4) We design a multiround iterative game (MRIG) algorithm based on arithmetic descent to solve the data offloading incentive problem and demonstrate our proposed design's superiorities through the numerical results. First, the proposed algorithm can be guaranteed to converge within 200 iterations. Second, the algorithm can significantly improve the mean value of USD in the system.

The remainder of this article is organized as follows. We first describe the system model in Section II. Sections III and IV address the task offloading destination problem and the data offloading incentive problem, respectively. The simulation results are discussed in Section V. Finally, we conclude this work in Section VI.

II. System Model

The network architecture of the multi-UAV-assisted MEC system considered in this article is shown in Fig. 1. The edge server deploys near the base station (BS) in the original MEC system and is generally called a BS-MEC server. In this article, we consider a scenario where the BS-MEC server is damaged due to natural disasters. The communication and computing capabilities of the BS-MEC server are lost and cannot be repaired in a short time period. To ensure the basic communication and computing needs of users within the coverage area of the damaged BS-MEC server, we introduce multiple identically configured UAVs that carry communication modules and computing modules. With the help of available ground gateways that are connected to the core network, signals are bridged from the neighboring BSs to the original coverage of the damaged BS via UAVs. The ground gateways mentioned here can be available communication facilities, such as neighboring BSs of the damaged BS, and emergency communication vehicles. The UAVs carry computing modules called UAV-MEC servers. These UAV-MEC servers will serve users in the original BS-server coverage area, sharing part of the load of the damaged BS-MEC servers. In this case, users within the original BS-MEC signal coverage area have two available types of data processing, as follows.

1) Offload to one of the UAV-MEC servers;
2) *Local Execution (LE):* As opposed to the UAV-MEC server, both the nearby gateway and the mobile device are considered local. We refer to offloading to the nearby gateways and computing on the mobile devices as LE.

We denote the set of introduced UAVs by $K = \{1, 2, \ldots, k\}$. We assume that the original BS coverage area contains a group of users (where each user is using one device) denoted by the set $N = \{1, 2, \ldots, n\}$ and that each user has a divisible computational task. We define the data set of users as $D = \{d_i\}$, $i \in N$, where d_i represents the amount of computing data of user i.

We first define the task offloading destination profile of the users as the vector $\boldsymbol{Z} = \{z_1, z_2, \ldots, z_n\}$, where $z_i \in K, i \in N$, denotes the task destination decision of user i and each computational task can only be offloaded to one UAV. After determining the task offloading destination, we refer to the vector $\boldsymbol{X} = \{x_1, x_2, \ldots, x_n\}$ as the data offloading decision profile of the users, where $x_i \in [0, d_i]$ denotes the data offloading strategy for user i, i.e., the amount of data that user i offloads to the specified UAV. Additionally, we denote the pricing decision profile for the UAV-MEC server as the vector $\boldsymbol{P} = \{p_1, p_2, \ldots, p_n\}$, where p_i represents the pricing of each unit of data offloaded by the UAV-MEC server for each user, i.e., the price that the UAV-MEC server charges user i for the computing resources required for a unit of data. Once user i offloads a unit of data to the specified UAV-MEC server, it pays p_i units of remuneration to that UAV-MEC server.

In this section, we introduce the measurement methods of the UAV-MEC server utility and user utility. For ease of reading, we summarize the main notation used in this article in Table I.

TABLE I
MAIN NOTATION USED IN THIS ARTICLE

Notation	Definition
N	$= \{1, 2, ..., n\}$, a group of users
d_i	the amount of computing data of user i, $i \in N$
K	$= \{1, 2, ..., k\}$, the set of introduced UAVs
$\boldsymbol{Z}$	$= \{z_1, z_2, ..., z_n\}$, the task offloading destination profile of users
$\boldsymbol{X}$	$= \{x_1, x_2, ..., x_n\}$, the data offloading decision profile of users
$\boldsymbol{P}$	$= \{p_1, p_2, ..., p_n\}$, the pricing decision profile of users
$f_{uav,j}$	the computing capacity of UAV-MEC server j, $j \in K$
$P^{com}_{uav,j}$	the CPU power of UAV-MEC server j
α	the task coding statement
$M_{uav,j}$	the number of tasks offloaded to UAV j
P_{hov}	the minimum hovering power of UAVs
η	the power efficiency
$U_{uav,j}$	the utility function of UAV-MEC server j
$U_{user,i}$	the utility function of user i
λ	the user's signal strength
σ^2	the initial SoU of the users before establishing the UAV-assisted MEC system
β	the adjustment parameter of the SoU
E	the total energy consumption of all UAVs in the system
ε	the maximum energy budget of the battery carried by the UAVs
$\boldsymbol{Z}^*$	the optimal task offloading destination profile
N_j	the set of users that plan to offload data to UAV j
$\boldsymbol{X}_j$	$= \{x_i\}, i \in N_j$, the data offloading decision policy used by each user in N_j to respond based on the specified pricing of UAV-MEC server j
$\boldsymbol{P}_j$	$= \{p_i\}, i \in N_j$, the pricing decision profile specified by UAV-MEC server j in the Stackelberg game
$\boldsymbol{X}^*_j$	$= \{x^*_i\}, i \in N_j$, the Nash equilibrium point of the Stackelberg game, i.e., the optimal data offloading decision profile
$\boldsymbol{P}^*_j$	the optimal pricing decision profile
$x^*_i(p_i)$	the optimal response of user i when p_i is specified
$p^{\min}_i$	the minimum threshold of p_i
$p^{\max}_i$	the maximum threshold of p_i
π_1	the number of iterations needed for the SSGT algorithm to reach the equilibrium state
π_2	the number of game rounds of the MRIG algorithm
ω	the game precision of the MRIG algorithm

A. UAV-MEC Server Utility

The utility of the UAV-MEC server is affected by many factors, such as the UAV battery capacity, flight trajectory [33], and flight altitude [34]. However, in this article, to study the incentive mechanism of resource pricing more intuitively, we only consider the battery capacity of UAVs, i.e., the energy consumption of UAVs in the offloading process.

We denote the computing capacity and the CPU power of the UAV-MEC server j by $f_{\text{uav},j}$ and $P^{\text{com}}_{\text{uav},j}$, respectively. Facing multiple simultaneous offloading tasks, UAV-MEC servers adopt the computational power allocation strategy of average distribution, with the main purpose of simplifying the computational model. This allocation strategy, although simple, is applicable to most MEC scenarios. Assume that the computational power of each UAV-MEC server is equally distributed to all tasks requesting offloading to that UAV and that $M_{\text{uav},j}$ denotes the number of tasks offloaded to UAV j at the same time. The computing time of UAV-MEC server j in executing the offloading data of user i can be expressed as

$$t^{\text{uav},j}_{i,\text{com}} = \frac{\alpha x_i}{f_{\text{uav},j}/M_{\text{uav},j}} = \frac{\alpha x_i \cdot M_{\text{uav},j}}{f_{\text{uav},j}} \tag{1}$$

where αx_i represents the number of CPU cycles required when the amount of offloading data is x_i, and the parameter α is a coefficient related to data encoding. Then, the computing energy consumption of UAV-MEC server j for user i is

$$\begin{aligned} E^{\text{uav},j}_{i,\text{com}} &= t^{\text{uav},j}_{i,\text{com}} \cdot P^{\text{com}}_{\text{uav},j} \\ &= \frac{\alpha x_i \cdot M_{\text{uav},j}}{f_{\text{uav},j}} \cdot P^{\text{com}}_{\text{uav},j}. \end{aligned} \tag{2}$$

Due to the limited energy resources of the UAV, in addition to the computing energy consumption generated by the UAV-MEC server in performing data offloading, we must consider the energy consumption of UAV hovering in the offloading process. According to [35], the energy consumption of UAV hovering can be expressed as $E_{\text{hov}} = P_{\text{hov}}/\eta$, where P_{hov}

denotes the minimum power of hovering and η denotes the power efficiency. Stolaroff et al. [35] discussed the energy consumption of UAV hovering and presented a detailed derivation of the formula. As noted by [35], we use the overall power efficiency of the UAV to correct for the theoretical minimum power. As several studies (e.g., [36] and [37]) have done, we also adopt the formula mentioned above. The specific derivation of the formula can be seen in [35], and we do not present more details in this article.

Then, if user i offloads some data to UAV-MEC server j, the energy consumption equation can be expressed as

$$\begin{aligned} E_i^{\text{uav},j} &= E_{i,\text{com}}^{\text{uav},j} + E_{\text{hov}} \\ &= \frac{\alpha x_i \cdot M_{\text{uav},j}}{f_{\text{uav},j}} \cdot P_{\text{uav},j}^{\text{com}} + \frac{P_{\text{hov}}}{\eta}. \end{aligned} \tag{3}$$

To more easily represent the offloading energy consumption of each UAV-MEC server in the system, we define the task destination decision function $C(z_i, \delta)$ for user i as

$$C(z_i, \delta) = \begin{cases} 1, \text{ if } z_i = \delta \\ 0, \text{ otherwise.} \end{cases} \tag{4}$$

Then, the total energy consumption equation of UAV j during the whole offloading process can be expressed as

$$E_{\text{uav},j} = \sum_{i=1}^{n} C(z_i, j) E_i^{\text{uav},j}. \tag{5}$$

Due to the incentive mechanism, the UAV-MEC server consumes energy in performing computational offloading and receives payments from users, the value of which depends on the amount of data offloaded by the user and the pricing set by the UAV. We measure the utility of the UAV-MEC server in terms of its revenue in computation offloading. The utility function of UAV-MEC server j is defined as the difference between the compensation obtained by its participation in computation offloading and its energy consumption. The equation is expressed as

$$\begin{aligned} U_{\text{uav},j} &= \sum_{i=1}^{n} C(z_i, j) p_i x_i - E_{\text{uav},j} \\ &= \sum_{i=1}^{n} C(z_i, j) \Big(p_i x_i - E_i^{\text{uav},j} \Big). \end{aligned} \tag{6}$$

B. User Utility

Compared to the energy resources of UAVs, those of mobile devices are quite sufficient. Therefore, energy consumption does not need to be considered when determining user utility. Since the focus of our study is on the offloading incentive mechanism, one of the factors that must be considered for user utility is the offloading cost. In addition, we consider the subjective evaluation of the user.

To better measure the subjective experience of users in performing computation offloading operations, we introduce the concept of the Satisfaction of User (SoU), an exponential function that can reflect users' computation offloading willingness. We use the value of SoU to measure the evaluation results intuitively. The SoU function is concave and quantitatively

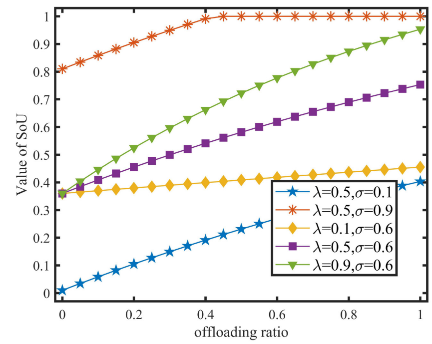

Fig. 2. SoU with different offloading ratios when parameters λ and σ have different values.

describes user satisfaction with the offloading activity according to the data offloading ratio at which users offload to the specified UAV-MEC server. The SoU function expression $s(x_i)$ for user i is defined as

$$s(x_i) = 1 - e^{-\lambda_i x_i / d_i} + \sigma_i^2 \quad \forall i \in N \tag{7}$$

where the parameter $\lambda_i \in [0, 1]$ denotes the signal strength when user i executes offloading and the expressions σ_i^2 and $\sigma_i \in [0, 1]$ denote the initial SoU of user i before establishing the UAV-assisted MEC system, the value of which depends on the computing and communication capabilities of the nearby gateway before UAVs are deployed and the user's own computing requirements. When the signal from the nearby gateway is very weak or user i is not in the communication range of the nearby gateway, σ_i^2 tends toward 0. Conversely, when the signal from the nearby gateway is very strong, the requirements of user i are lower, and the value of σ_i^2 is larger. Additionally, considering the practical implications of the SoU function representation, the value range of the function is limited from 0 to 1, i.e., $s(x_i) \in [0, 1]$.

Some of the properties of the SoU function are shown in Fig. 2. Fig. 2 shows the SoU with different offloading ratios when parameters λ and σ take different values. Obviously, the SoU function value can reach the maximum even without the user offloading all its computing data to the UAV-MEC server when λ and σ are both large. In this case, offloading partial data can fully meet the user computing requirements. At the same time, we observe that the SoU function value improves very slowly with increasing offloading ratio when the value of λ is small. In other words, when the user's signal is poor, even if a large amount of data are offloaded to the UAV-MEC server, the SoU function value will not be significantly improved. This indicates that users are not encouraged to execute computation offloading when their signal strengths are poor.

Next, we discuss the utility of users, which considers both the SoU and offloading cost components. In the UAV-assisted MEC system, if users intend to offload data to the specified UAV-MEC server, they must pay the corresponding offloading cost. In addition, we set the parameter β as the adjustment factor of the SoU function to solve the problem of determining the numerical difference between the SoU function value and the offloading cost. The value of β can reflect the emphasis on user experience during the offloading process. For user i, its utility is defined as the difference between the adjusted SoU and the offloading cost, which can be expressed as

$$\begin{aligned} U_{\text{user},i} &= \beta s(x_i) - p_i x_i \\ &= \beta\left(1+\sigma_i^2\right) - \beta e^{-\lambda_i x_i / d_i} - p_i x_i. \end{aligned} \tag{8}$$

III. Task Offloading Destination

To complete the computation offloading process, we first need to determine the task offloading destination of each user. This is a preoffloading simulation process, with no actual offloading operation. The first purpose of preoffloading is to initially complete the allocation of computing power to the UAV-MEC servers through the process of determining where tasks are offloaded to. On the other hand, it is to test the data carrying capacity of each UAV-MEC server, so that the whole data offloading operation can be completed properly even in the extreme case that all users prefer to pay a huge price to offload all the data to the UAV-MEC server the actual data offloading process. In this section, we first establish the task offloading destination model and then design the SSGT algorithm to solve the problem.

A. Task Offloading Destination Model

In this section, we first assume that each computational task is completely offloaded, i.e., $x_i = d_i \quad \forall i \in N$, and that no pricing is applied to the computing resources of the UAVs. We search for the optimal task offloading destination strategy that can minimize the total energy consumption of all UAVs in the system.

According to (5), the total energy consumption of all UAVs in the task offloading destination profile $\mathbf{Z}$ can be expressed as

$$\begin{aligned} E(\mathbf{Z}) &= \sum_{j=1}^{k} E_{\text{uav},j} \\ &= \sum_{j=1}^{k}\sum_{i=1}^{n} C(z_i, j) E_i^{\text{uav},j}. \end{aligned} \tag{9}$$

Based on the energy consumption model shown above, the problem of determining the task offloading destination can be described as follows:

$$\begin{aligned} \min\ & \sum_{j=1}^{k}\sum_{i=1}^{n} C(z_i, j) E_i^{\text{uav},j} \\ \text{s.t.}\ & \text{C1}: z_i \in K \quad \forall i \in N \\ & \text{C2}: \sum_{j=1}^{k} C(z_i, j) = 1 \quad \forall i \in N \\ & \text{C3}: E_{\text{uav},j} \le \varepsilon \quad \forall j \in K. \end{aligned} \tag{10}$$

In (10), C1 represents the possible value of the destination decision of each task as the UAV's serial number. Constraint C2 indicates that there can be only one offloading decision per task. Constraint C3 means that the energy consumption of each UAV in the whole offloading process must be lower than a certain fixed threshold ε, and the value of ε is the battery energy budget of each UAV.

B. Server Selection Game-Theoretic Algorithm

In this section, we introduce a general tool to solve the above problem: game theory. In the system, each user acts as an individual, and their interactions directly affect the final optimal solution of the problem. These users, as decision makers, need to work together to achieve the system goals. Each decision maker wants to obtain the most beneficial solution for itself. However, at the same time, the decision makers will constrain each other, which creates a situation where each decision maker plays against the others. In the whole game process, no decision maker is allowed to have the motivation of self-interest only. To address the decisions of the other decision makers, each decision maker will implement the most beneficial decision in the current environment. Ultimately, the result is a mutually satisfactory solution for all decision makers.

We view the interaction between users in the process of obtaining the task offloading destination as a game process. We determine the three elements of the game: 1) the set of participants; 2) the set of strategies of each participant; and 3) the utility function of each participant. The set of participants in the game is the set of users, $i \in N, N = \{1, 2, \ldots, n\}$; the strategy space of each participant is denoted as $z_i, i \in N$. Equation (9) is expanded as follows:

$$\begin{aligned} E(Z) &= E_1(z_1, z_{-1}) + \cdots + E_i(z_i, z_{-i}) + \cdots \\ &\quad + E_n(z_n, z_{-n}) \\ &= \sum_{j=1}^{k} C(z_1, j) E_1^{\text{uav},j} + \cdots + \sum_{j=1}^{k} C(z_i, j) E_i^{\text{uav},j} \\ &\quad + \cdots + \sum_{j=1}^{k} C(z_n, j) E_n^{\text{uav},j} \\ &= C(z_1, 1) E_1^{\text{uav},1} + \cdots + C(z_1, j) E_1^{\text{uav},j} + \cdots \\ &\quad + C(z_1, k) E_1^{\text{uav},k} + \cdots + C(z_i, 1) E_i^{\text{uav},1} + \cdots \\ &\quad + C(z_i, j) E_i^{\text{uav},j} + \cdots + C(z_i, k) E_i^{\text{uav},k} + \cdots \\ &\quad + C(z_n, 1) E_n^{\text{uav},1} + \cdots + C(z_n, j) E_n^{\text{uav},j} + \cdots \\ &\quad + C(z_n, k) E_n^{\text{uav},k} \end{aligned}$$

where $E_i(z_i, z_{-i})$ denotes the utility function of user i and $z_{-i} = \{z_1, z_2, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n\}$ denotes the set of strategies of users other than user i. Combining the features of (5), the possible values of $E_i(z_i, z_{-i})$ are expressed as

$$E_i(z_i, z_{-i}) = \begin{cases} E_i^{\text{uav},1}, & \text{if } z_i = 1 \\ E_i^{\text{uav},2}, & \text{if } z_i = 2 \\ ,\ldots & \\ E_i^{\text{uav},j}, & \text{if } z_i = j \\ ,\ldots & \\ E_i^{\text{uav},k}, & \text{if } z_i = k. \end{cases} \tag{11}$$

In summary, the strategic formulation of the unloaded game is denoted as $\Psi = \{1, 2, \ldots, i, \ldots, n;\ z_1, z_2, \ldots, z_i, \ldots, z_n;\ E_1, E_2, \ldots, E_i, \ldots, E_n\}$.

The next question of interest is whether there is a Nash equilibrium in the game. Two relevant lemmas [38] are given as follows.

Algorithm 1 SSGT Algorithm

Input: N, K, D, $f_{uav,k}$, $P^{com}_{uav,k}$, P_{hov}, η, α.
Output: the destination decision profile $\boldsymbol{Z}^*$ and the minimum total energy consumption of all UAVs $E_{\min}$.

```
Initialization: each user randomly selects the destination strategy z_i, i ∈ N and records the corresponding decision as Z*;
Calculate the initial value of E*;
repeat
    Get the current network status;
    Each user competes for the strategy update opportunity;
    if user l, l ∈ N obtains the opportunity, then
        Calculate the corresponding E' of its other destination options, respectively;
        if ∃E' < E*, then
            Update z_l in Z* and set E* = E';
        else
            Keep z_l unchanged;
        end if
    else
        Keep z_l unchanged;
    end if
until an equilibrium is achieved;
Output Z* and E_min = E*.
```

Lemma 1: Any finite game has at least one Nash equilibrium.

Lemma 2: A game is considered to be finite if it satisfies the following two conditions.

1) The game has a finite number of participants.
2) There are a finite number of strategies available to each participant.

Theorem 1: Game $\Psi = \{1, 2, \dots, i, \dots, n;\ z_1, z_2, \dots, z_i, \dots, z_n;\ E_1, E_2, \dots, E_i, \dots, E_n\}$ is a finite game, and it has a Nash equilibrium.

Proof: If game Ψ satisfies the conditions of a finite game, it is a finite game. The participants in game Ψ are the users in the system, and the number of users is finite, so condition 1 of the finite game is satisfied. Each participant can choose any UAV in the system as a strategy, and the number of drones is finite, so condition 2 of the finite game is satisfied. By Lemma 2, game Ψ is a finite game. Therefore, by Lemma 1, a Nash equilibrium exists for game Ψ. ■

Therefore, we propose the SSGT algorithm based on game theory to solve the problem of task offloading destinations. The details of which are described in Algorithm 1.

In the SSGT algorithm, the task offloading destination strategy is first initialized by random selection. Then, all users compete for the opportunity to update the destination decision. If a user wins the update opportunity and its other destination choices yield a smaller total energy consumption for all UAVs, the update step is executed. After several iterations, when the current strategy is no longer updated, the equilibrium state is reached. The minimum total energy consumption value E^* and the optimal task offloading destination profile $\boldsymbol{Z}^*$ are obtained. The whole algorithm uses one layer of loops, and the time complexity is denoted as $O(\pi_1)$, where π_1 is the number of iterations needed to reach the equilibrium state.

IV. Data Offloading Incentive

After determining the offloading destinations of each task through the preoffloading action in Section III, we introduce the data offloading incentive to study the actual data offloading problem in this section. We specify a UAV j and the users who plan to offload to this UAV, where the set of users is denoted as $N_j, N_j \subseteq N$. We first model the data offloading incentive problem as a Stackelberg game process and then prove the existence of a Nash equilibrium between the UAV-MEC server and users in our proposed Stackelberg game model. Finally, we propose an iterative algorithm based on arithmetic descent to solve the Nash equilibrium, which is the final solution of the game.

A. Stackelberg Game Model

We describe the interaction between the specified UAV-MEC server and the users that plan to offload to this UAV as a two-stage Stackelberg game to obtain the optimal pricing strategy and optimal data offloading strategy, where the UAV-MEC server is the leader and the users are the followers. In the first stage, the UAV-MEC server specifies a pricing decision profile $\boldsymbol{P}_j = \{p_i\}, i \in N_j$. In the second stage, each user responds to its data offloading strategy based on the specified pricing. The data offloading decision profile is expressed as $\boldsymbol{X}_j = \{x_i\}, i \in N_j$, where $x_i = f(p_i)$.

Under the condition that the UAV-MEC server specifies the pricing strategy in the first stage, the optimal data offloading strategy problem for user i is as follows:

$$\begin{aligned} \max U_{\text{user},i} \quad & (x_i) \\ \text{s.t.} \quad & \text{C1} : \forall i \in N \\ & \text{C2} : C(z_i, j) = 1 \\ & \text{C3} : x_i \in [0, d_i]. \end{aligned} \tag{12}$$

In (12), $U_{\text{user},i}(x_i)$ is the utility of user i in (8). Constraints C1 and C2 indicate that user i is within the signal coverage of UAV j and that the task offloading destination of user i is UAV j; satisfying C1 and C2 is equivalent to $i \in N_j$. Constraint C3 means that the amount of data offloaded to the UAV-MEC server is between 0 and the maximum amount of computing data for user i.

The UAV-MEC server first predicts how each user in the second stage will respond to the specified pricing strategy. Then, we study the optimal decision of the UAV-MEC server in the first stage based on this prediction. The optimal pricing strategy problem of the UAV-MEC server can be expressed as follows:

$$\begin{aligned} \max U_{\text{uav},j} \quad & \left(\boldsymbol{X}_j, \boldsymbol{P}_j\right) \\ \text{s.t.} \quad & \text{C1} : p_i > 0 \quad \forall i \in N_j \\ & \text{C2} : x_i(p_i) \le d_i \quad \forall i \in N_j \\ & \text{C3} : E_{\text{uav},j} \le \varepsilon. \end{aligned} \tag{13}$$

In (13), $U_{\text{uav},j}(\boldsymbol{X}_j, \boldsymbol{P}_j)$ is the utility of UAV-MEC server j in (6). Constraint C1 states that the pricing of each user by the UAV-MEC server must be positive. Constraint C2 means that the amount of offloading data for each user response is limited and that the maximum response cannot exceed the amount of task data. Constraint C3 is the same as C3 in (10), which is intended to account for the energy budget of the UAV's battery.

B. Iteration-Based Arithmetic Descent Solution

Before solving the data offloading incentive problem, we first prove the existence of the Nash equilibrium of the Stackelberg game in this section.

When the game reaches the Nash equilibrium, the utility of each participant in the game can reach the maximum, and a unilateral change in one participant's strategy will not increase its utility. First, a well-known lemma [39] is presented follows.

Lemma 3: At least one Nash equilibrium for a noncooperative game exists if, for $\forall i \in N$.

1) The strategy space $\mathbf{X}$ is a nonempty, convex, and compact subset of some Euclidean space.
2) The utility function $U_{\text{user},i}(x_i)$ is continuous and quasi-convex in $\mathbf{X}$.

Definition 1: For users, the data offloading decision profile $\boldsymbol{X}_j^* = \{x_i^*\}, i \in N_j$ is the Nash equilibrium of the Stackelberg game. If and only if the Nash equilibrium point $\boldsymbol{X}_j^*$ is reached, no user can increase its utility by changing its data offloading strategy; i.e., for any user i

$$U_{\text{user},i}\left(x_i^*\right) \geq U_{\text{user},i}(x_i), x_i \in \boldsymbol{X}_j \quad \forall i \in N_j.$$

Theorem 2: The specified pricing decision profile of the UAV-MEC server is $\mathbf{P}_j = \{p_i\}, i \in N_j$. Under this pricing decision profile, each user plays a noncooperative game, and the game in which the utility function of user i is $U_{\text{user},i}(x_i)$ has the Nash equilibrium solution.

Proof: Clearly, the data offloading decision profile $\boldsymbol{X}_j$ of all users is a bounded nonempty closed convex set in Euclidean space. It is known that bounded closed sets in Euclidean spaces are compact. In addition, the utility function $U_{\text{user},i}(x_i)$ is continuous in its strategy space.

The following items prove that the utility function meets the characteristics of a concave function. The first derivative of the utility function $U_{\text{user},i}(x_i)$ of user i with respect to x_i can be obtained as

$$\frac{\partial U_{\text{user},i}}{\partial x_i} = \frac{\beta\lambda_i}{d_i} e^{-\lambda_i x_i / d_i} - p_i. \tag{14}$$

The second derivative of the utility function $U_{\text{user},i}(x_i)$ of user i with respect to x_i can be obtained as

$$\frac{\partial^2 U_{\text{user},i}}{\partial x_i^2} = -\frac{\beta\lambda_i^2}{d_i^2} e^{-\lambda_i x_i / d_i}. \tag{15}$$

Because $\lambda_i > 0, d_i > 0$, we can obtain that the second derivative is less than zero. The utility function $U_{\text{user},i}(x_i)$ of user i is strictly concave. Therefore, the Nash equilibrium solution exists. ■

The existence of the Nash equilibrium proves that there is always a Nash equilibrium point in the data offloading strategy of users for a specified pricing strategy of the UAV-MEC server. We can always find an optimal pricing decision profile and the corresponding data offloading decision profile to maximize the utility of the UAV-MEC server and users. This shows that the Stackelberg game has a subgame perfect Nash equilibrium.

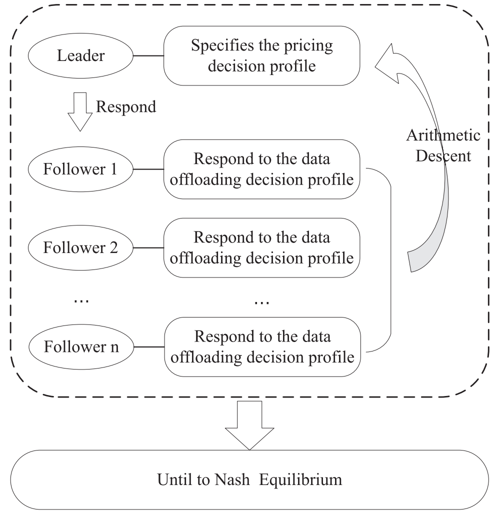

Fig. 3. MRIG algorithm mechanism design.

Then, we discuss a fundamental question: how do users, as followers, respond to the optimal data offloading strategy according to the pricing strategy of the UAV-MEC server? For the pricing strategy specified for the UAV-MEC server, by setting (14) to zero, we obtain the optimal strategy for user i

$$x_i^*(p_i) = -\frac{d_i}{\lambda_i} \ln \frac{d_i p_i}{\beta\lambda_i}. \tag{16}$$

Since the first derivative of $x_i^*(p_i)$ is negative, $x_i^*(p_i)$ is a monotonically decreasing function. When the computing resource pricing of the UAV-MEC server increases, the amount of data offloaded by user i to the UAV-MEC server decreases. Because $x_i \in [0, d_i]$, it is easy to see that p_i has a threshold, which can be expressed as

$$p_i^{\max} = \frac{\beta\lambda_i}{d_i} \tag{17}$$

$$p_i^{\min} = \frac{\beta\lambda_i}{d_i} e^{-\lambda_i}. \tag{18}$$

Therefore, given the pricing of the UAV-MEC server, the optimal response of user i is expressed as

$$x_i^*(p_i) = \begin{cases} 0, & p_i \geq p_i^{\max} \\ -\frac{d_i}{\lambda_i} \ln \frac{d_i p_i}{\beta\lambda_i}, & p_i^{\min} < p_i < p_i^{\max} \\ d_i, & 0 < p_i \leq p_i^{\min}. \end{cases} \tag{19}$$

We propose the MRIG algorithm to solve the Nash equilibrium as Algorithm 2. This algorithm uses iteration to play multiple rounds of the game. The mechanism design of the algorithm is shown in Fig. 3.

In the MRIG algorithm, the number of iterations of the game increases progressively with each round. In each game round, the UAV-MEC server will first assign the pricing strategy as the leader. We set the initial pricing decision profile as the maximum pricing strategy for each user. Once the iteration

Algorithm 2 MRIG Algorithm

Input: N_j, D, $f_{uav,j}$, $P^{com}_{uav,j}$, P_{hov}, η, β, α, ε, ω.
Output: an optimal pricing profile $\boldsymbol{P}^*_j$ and an optimal data offloading decision profile $\boldsymbol{X}^*_j$.

```
Calculate the UAV-MEC server pricing thresholds P_j^max and P_j^min for each user;
Set the initial number of iterations;
repeat
    Calculate the range of descent of the pricing strategy for each user in the current round game;
    Set P_j^* = {p_i^max}, ∀i ∈ N_j as initial pricing decision profile for the UAV-MEC server j;
    repeat
        Each user competes for the opportunity to have the UAV-MEC server update the corresponding pricing strategy;
        if user i wins the strategy update opportunity, then
            Update p_i in P_j^*;
            User i makes the optimal response to the UAV -MEC server;
            Calculate the current utility of the UAV-MEC server j;
            Judge whether to update the current local optimal solution P_j^*;
        else
            Keep p_i unchanged;
        end if
    until each user completes the possible updates;
    Obtain the local optimal solution P_j^* for this round of the game;
    Increase the number of iterations;
until a Nash equilibrium is obtained;
Output P_j^* and X_j^* = {x_i^*(p_i^*)}.
```

begins, each user will compete for the opportunity to have the UAV-MEC server update the corresponding pricing strategy. After user i wins the update opportunity, it will carry out the update steps. The pricing strategy is updated by employing arithmetic descent, and the decreasing range of arithmetic descent is determined by the number of game rounds and the game precision of the round.

After the update, user i will make the optimal response according to (19) and determine whether it needs to update the current local optimal solution according to the utility of the UAV-MEC server at this time. When each user completes the possible update, this round of the game ends. After each game round, we obtain a local optimal solution. Then, the game will enter the next round. When the optimal utility of the UAV-MEC no longer increases, that is, all users perform the optimal response, the whole system will reach a Nash equilibrium. Let π_2 denote the number of game rounds, and let ω denote the game precision. The MRIG algorithm includes two levels of loops, where the number of executions of the outer loop is π_2 and the number of executions of the inner loop is π_2/ω. Then, the number of computations is $\pi_2 * (\pi_2/\omega) = {\pi_2}^2/\omega$, so the time complexity of the algorithm is $O({\pi_2}^2)$.

TABLE II
VALUES OF SOME OF THE SIMULATION DATA

Notation	Value
d_i	randomly selected from [10,150] MB
α	1900 cycles/byte
$P^{com}_{uav,j}$	randomly selected from [0.1,0.15] W
$f_{uav,j}$	randomly selected from $[1 \times 10^9, 5 \times 10^9]$ cycles/s

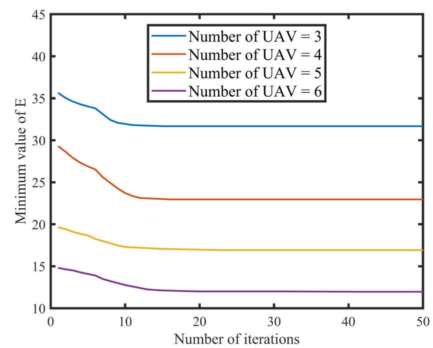

Fig. 4. Variation in the number of offloading tasks per UAV as the number of iterations increases.

V. Simulation Results

In this section, we carry out the simulation experiments using MATLAB. The first step is the task offloading destination simulation, and the second step is the data offloading incentive simulation.

According to [40], the computing capacity of the UAV-MEC server is a random value between 1×10^9 and 5×10^9cycles/s. The CPU power of the UAV-MEC server is a random value between 0.1 and 0.5 W. We consider ×264 CBR encoding applications, where a data unit requires a CPU revolution of 1900 cycles/byte, i.e., $\alpha = 1900$ cycles/byte. The amount of data for each task is randomly distributed in the range [10, 150] MB. Some parameters required for the simulation experiments are summarized in Table II.

In the first step, we first perform the task offloading destination simulation by implementing the SSGT algorithm, assuming a random distribution of $n = 20$ users in the signal coverage area.

Fig. 4 shows the variation in the minimum value of E under different numbers of UAVs as the number of iterations increases. In Fig. 4, the mean filtering method is used to smooth the curve. For different number of UAVs, the minimum value of E decreases with each iteration and can ultimately converge within a few tens of iterations. Meanwhile, we can observe that the greater the number of UAVs is, the smaller the minimum value of E to which the algorithm eventually converges. That is, an increase in the number of task offloading destinations of users will reduce the total energy consumption of all UAVs in the system.

Fig. 5 shows the variation in the number of tasks offloaded to each UAV as the number of iterations increases when $k = 4$. In Fig. 5, we observe that as the number of iterations increases, the number of tasks offloaded to each UAV gradually tends to become balanced. Through the simulation steps, we can

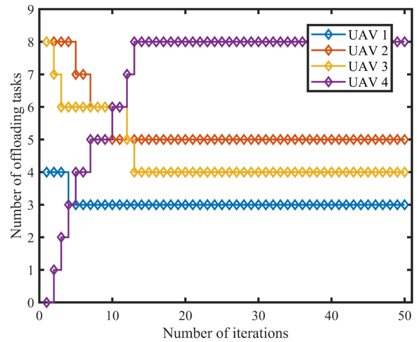

Fig. 5. Variation in the value of E under different numbers of UAVs as the number of iterations increases.

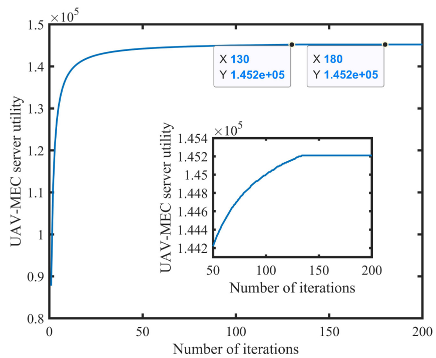

Fig. 6. Variation in the UAV-MEC server utility as the number of iterations increases.

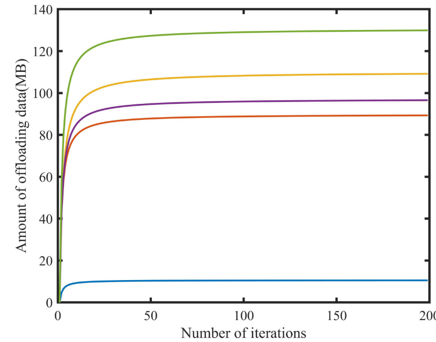

Fig. 7. Variation in the amount of offloading data for each user as the number of iterations increases.

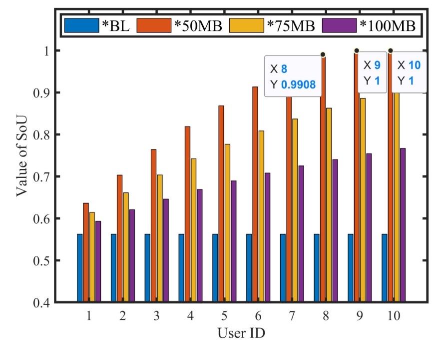

Fig. 8. SoU variation corresponding to each user for different total numbers of computations.

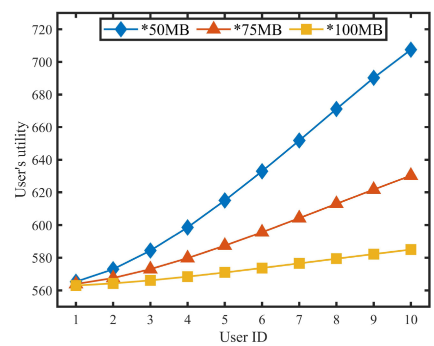

Fig. 9. User utility variation corresponding to each user for different total numbers of computations.

determine the offloading correspondence between each UAV and each task for the next specific offloading operation.

In the second step, we examine the incentive mechanism for a particular UAV j and the users who plan to offload data to this UAV. We perform simulation experiments on the data offloading incentive mechanism by implementing the MRIG algorithm.

First, to verify the existence of the Nash equilibrium, we assume that the set N_j contains five users, i.e., there are five users that plan to offload data to UAV j in the system. Fig. 6 shows the variation in the UAV-MEC server utility as the number of iterations increases. As the iteration continues, the UAV-MEC server utility continues to grow and then gradually tends to become stable until it converges to an equilibrium state. At this point, the game reaches the Nash equilibrium state, i.e., the utility of the UAV-MEC server reaches the maximum value, which demonstrates the effectiveness of the Stackelberg game from an experimental perspective. Fig. 7 shows the variation in the amount of offloaded data for each user. As the iteration continues, the amount of offloaded data for each user also continues to grow and gradually tends to become stable. When the user's offloading strategies become stable, the maximum user utility is reached, and the optimal offloading state for all users is reached.

Next, we evaluate the impact of the user signal strength and the amount of computing data on the value of SoU and the user's utility. To facilitate the calculation of the total amount of computation in the coverage area, we assume that N_j contains ten users with the same amount of data for the experiments. We number each user from 1 to 10 and record the number as the User ID. The value of the parameter $\lambda$is set to $0.1, 0.2, \ldots, 0.9, 1$ according to the User ID. The user with ID 1 has the lowest signal strength, and the user with ID 10 has the highest signal strength. We set the user's amount of computing data to 50, 75, and 100 MB for the experiments. Figs. 8 and 9 show the SoU and user utility variation corresponding to each user for different total numbers of computations in the signal coverage area, respectively.

In Fig. 8, to observe the improvement effect of the SoU function after introducing the UAV-MEC server, we also obtain the SoU function value before introducing the UAV-MEC server as a baseline (identified as BL in Fig. 8). From Fig. 8, we can observe that the SoU function value is always largest when the amount of computing data is 50 MB in the face of different signal strengths. This indicates that the smaller the total amount of data in the system is, the greater the improvement in the SoU function value. In addition, Fig. 8 demonstrates that the SoU value gradually increases as the signal strength increases and increases fastest when the user's data amount is small. This indicates that the higher the signal strength of the user, the faster the SoU value increases. When the user's signal strength is large but the amount of computing data is small, the SoU value may reach 1. This indicates that the introduction of the UAV-MEC server has a significant improvement effect on the SoU metrics of users, especially when the user signal strength is large or the amount of computing data is small. This intuitively proves that it is indispensable to introduce the UAV-MEC server from the users' perspective.

In Fig. 9, we can observe that the user's utility gradually increases with signal strength, and the rate of increase becomes progressively faster. From the proof analysis process in Section IV, the pricing strategy specified by the UAV-MEC server is one of the determinants of the user's data offloading strategy in the game. The user's data offloading strategy also has a direct impact on the pricing strategy of the UAV-MEC server. Therefore, we analyze that the possible reasons for this change are the decrease in pricing and the slowing growth rate of the amount of offloading data, which reduces the user's offloading cost. This indicates that from the user's perspective, the larger the signal strength of the user is, the more it is encouraged to offload to the UAV-MEC server, which is consistent with the conclusion indicated in Fig. 2.

In addition, from Figs. 8 and 9, we can observe that when the signal strength of one user remains unchanged and the total amount of data in the coverage area increases, both its SoU and utility will decrease. This indicates that the SoU and utility of the user are affected not only by its factors but also by other users in the area.

Finally, to verify the performance of the MRIG scheme presented in Section IV, we introduce three additional computational offloading schemes as baselines.

1) *Offloading Scheme Based on Random Selection (OSRS) Scheme:* The amount of offloaded data is randomly selected from $[0, d_i]$.
2) *Offloading Scheme Based on Even Allocation (OSEA) Scheme:* Each user offloads the same percentage of data.
3) *LE Scheme:* All data are executed locally.

We conduct experiments for different numbers of users. Figs. 10 and 11 compare the mean value of SoU for all users and the UAV-MEC server utility under the four different offloading schemes, respectively. The advantages of the MRIG scheme can be seen in Figs. 10 and 11. The MRIG scheme significantly improves the mean value of SoU and the UAV-MEC server utility compared to the OSRS and OSEA. At the same time, by comparing it with the LE scheme, it further proves the necessity of introducing UAVs to assist computation.

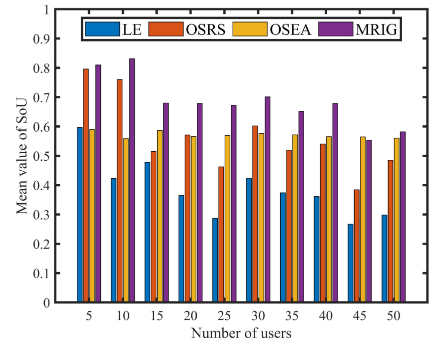

Fig. 10. Comparison of the mean value of SoU for all users under four different offloading schemes.

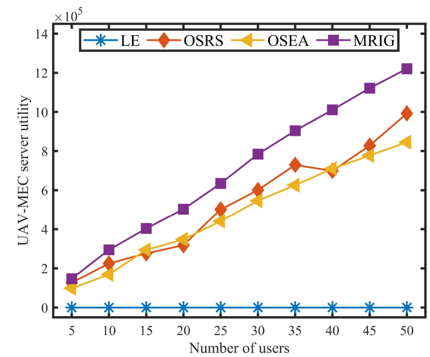

Fig. 11. Comparison of UAV-MEC server utility under four different offloading schemes.

VI. Conclusion

In this article, we first simulate the task offloading destination by preoffloading each computational task in a multi-UAV-assisted MEC system with the SSGT algorithm. Then, the offloading incentive mechanism is introduced to encourage the UAV-MEC server to participate in computation offloading, and the concept of SoU is proposed to measure users' offloading willingness. The computation offloading problem between a single UAV and users is described as a Stackelberg game. It is proven that the game can achieve a unique Nash equilibrium through game theory, and we design an iterative algorithm to obtain the Nash equilibrium strategy in the system. Finally, we compute the value of SoU, the UAV-MEC utility value, and the user's utility as the evaluation metrics to conduct simulations, which verify the superiority of our proposed MRIG algorithm.

References

[1] D. E. Culler, "The once and future Internet of Everything," *Mobile Comput. Commun. Rev.*, vol. 20, no. 3, pp. 5–11, 2016.
[2] B. Singh, S. Dhawan, A. Arora, and A. Patail, "A view of cloud computing," *Int. J. Comput. Technol.*, vol. 53, no. 4, pp. 50–58, 2013.
[3] A. Khan, M. Othman, S. A. Madani, and S. U. Khan, "A survey of mobile cloud computing application models," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 393–413, 1st Quart., 2014.
[4] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1657–1681, 3rd Quart., 2017.
[5] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.
[6] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2624–2661, 4th Quart., 2016.

[7] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 36–42, May 2016.
[8] N. Zhao et al., "UAV-assisted emergency networks in disasters," *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 45–51, Feb. 2019.
[9] F. Zhou, Y. Wu, H. Sun, and Z. Chu, "UAV-enabled mobile edge computing: Offloading optimization and trajectory design," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2018, pp. 1–6.
[10] S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2049–2063, Mar. 2018.
[11] N. Cheng et al., "Air-ground integrated mobile edge networks: Architecture, challenges, and opportunities," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 26–32, Aug. 2018.
[12] M. Dong, K. Ota, M. Lin, Z. Tang, S. Du, and H. Zhu, "UAV-assisted data gathering in wireless sensor networks," *J. Supercomput.*, vol. 70, no. 3, pp. 1142–1155, 2014.
[13] Z. Liu, X. Tan, M. Wen, S. Wang, and C. Liang, "An energy-efficient selection mechanism of relay and edge computing in UAV-assisted cellular networks," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 3, pp. 1306–1318, Sep. 2021.
[14] A. M. Seid, G. O. Boateng, S. Anokye, T. Kwantwi, G. Sun, and G. Liu, "Collaborative computation offloading and resource allocation in multi-UAV-assisted IoT networks: A deep reinforcement learning approach," *IEEE Internet Things J.*, vol. 8, no. 15, pp. 12203–12218, Aug. 2021.
[15] N. N. Ei, M. Alsenwi, Y. K. Tun, Z. Han, and C. S. Hong, "Energy-efficient resource allocation in multi-UAV-assisted two-stage edge computing for beyond 5G networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16421–16432, Sep. 2022.
[16] Y. Du, K. Yang, K. Wang, G. Zhang, and D. Chen, "Joint resources and workflow scheduling in UAV-enabled wirelessly-powered MEC for IoT systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 10187–10200, Oct. 2019.
[17] Y. Xu, T. Zhang, J. Loo, D. Yang, and L. Xiao, "Completion time minimization for UAV-assisted mobile-edge computing systems," *IEEE Trans. Veh. Technol.*, vol. 70, no. 11, pp. 12253–12259, Nov. 2021.
[18] X. Zhang, Y. Wang, G. Geng, and J. Yu, "Delay-optimized multicast tree packing in software-defined networks," *IEEE Trans. Services Comput.*, vol. 16, no. 1, pp. 261–275, Jan./Feb. 2023.
[19] X. Zhang and T. Wang, "Elastic and reliable bandwidth reservation based on distributed traffic monitoring and control," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 12, pp. 4563–4580, Dec. 2022.
[20] Y. Xu, T. Zhang, Y. Liu, D. Yang, L. Xiao, and M. Tao, "UAV-assisted MEC networks with aerial and ground cooperation," *IEEE Trans. Wireless Commun.*, vol. 20, no. 12, pp. 7712–7727, Dec. 2021.
[21] Y. Wang, W. Fang, Y. Ding, and N. Xiong, "Computation offloading optimization for UAV-assisted mobile edge computing: A deep deterministic policy gradient approach," *Wireless Netw.*, vol. 27, no. 4, pp. 2991–3006, 2021.
[22] X. Gu, G. Zhang, M. Wang, W. Duan, M. Wen, and P.-H. Ho, "UAV-aided energy-efficient edge computing networks: Security offloading optimization," *IEEE Internet Things J.*, vol. 9, no. 6, pp. 4245–4258, Mar. 2022.
[23] J. Zhang, M. Li, Z. Chen, and B. Lin, "Computation offloading for object-oriented applications in a UAV-based edge-cloud environment," *J. Supercomputing*, vol. 78, no. 8, pp. 10829–10853, 2022.
[24] L. Yang, C. Xu, Y. Zhan, Z. Liu, J. Guan, and H. Zhang, "Incentive mechanism for computation offloading using edge computing: A Stackelberg game approach," *Comput. Netw.*, vol. 129, no. 24, pp. 399–409, 2017.
[25] K. Xie et al., "Distributed multi-dimensional pricing for efficient application offloading in mobile cloud computing," *IEEE Trans. Services Comput.*, vol. 12, no. 6, pp. 925–940, Nov./Dec. 2019.
[26] R. Myerson, *Game Theory: Analysis of Conflict*. Cambridge, MA, USA: Harvard Univ. Press, 1991.
[27] G. Mitsis, E. E. Tsiropoulou, and S. Papavassiliou, "Data offloading in UAV-assisted multi-access edge computing systems: A resource-based pricing and user risk-awareness approach," *Sensors*, vol. 20, no. 8, p. 2434, 2020.
[28] W. Fan, L. Yao, J. Han, F. Wu, and Y. Liu, "Game-based multitype task offloading among mobile-edge-computing-enabled base stations," *IEEE Internet Things J.*, vol. 8, no. 24, pp. 17691–17704, Dec. 2021.
[29] T. Fang, F. Yuan, L. Ao, and J. Chen, "Joint task offloading, D2D pairing, and resource allocation in device-enhanced MEC: A potential game approach," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3226–3237, Mar. 2022.
[30] Q. Luo, C. Li, T. H. Luan, W. Shi, and W. Wu, "Self-learning based computation offloading for Internet of Vehicles: Model and algorithm," *IEEE Trans. Wireless Commun.*, vol. 20, no. 9, pp. 5913–5925, Sep. 2021.
[31] R. Zheng, H. Wang, M. De Mari, M. Cui, X. Chu, and T. Q. S. Quek, "Dynamic computation offloading in ultra-dense networks based on mean field games," *IEEE Trans. Wireless Commun.*, vol. 20, no. 10, pp. 6551–6565, Oct. 2021.
[32] M. Chen, W. Liu, T. Wang, S. Zhang, and A. Liu, "A game-based deep reinforcement learning approach for energy-efficient computation in MEC systems," *Knowl.-Based Syst.*, vol. 235, Jan. 2022, Art. no. 107660.
[33] J. Zhang et al., "Stochastic computation offloading and trajectory scheduling for UAV-assisted mobile edge computing," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3688–3699, Apr. 2019.
[34] S. Kumar, S. Suman, and S. De, "Backhaul and delay-aware placement of UAV-enabled base station," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2018, pp. 634–639.
[35] J. K. Stolaroff, C. Samaras, E. R. O'Neill, A. Lubers, A. S. Mitchell, and D. Ceperley, "Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery," *Nat. Commun.*, vol. 9, pp. 1–13, Feb. 2018.
[36] K. Zhang, X. Gui, D. Ren, and D. Li, "Energy–latency tradeoff for computation offloading in UAV-assisted multiaccess edge computing system," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6709–6719, Apr. 2021.
[37] K. Kim and C. S. Hong, "Optimal task-UAV-edge matching for computation offloading in UAV assisted mobile edge computing," in *Proc. 20th Asia–Pacific Netw. Oper. Manage. Symp. (APNOMS)*, 2019, pp. 1–4.
[38] J. F. Nash, "Equilibrium points in n-person games," *Proc. Nat. Acad. Sci.*, vol. 36, no. 1, pp. 48–49, 1950.
[39] A. B. Mackenzie and L. A. Dasilva, *Game Theory for Wireless Engineers* (Synthesis Lectures on Communications). San Rafael, CA, USA: Morgan and Claypool Publ., 2006.
[40] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proc. USENIX Assoc.*, vol. 10, 2010, p. 4.