IEEE Open Journal of the Computer Society

Received 9 April 2021; accepted 23 July 2021. Date of publication 30 July 2021; date of current version 26 August 2021. The review of this paper was arranged by Guest Editors Sahil Garg, Alagan Anpalagan, Shree Krishna Sharma, Li-Chun Wang, Muhammad Zeeshan Shakir, Ning Zhang, and Linglong Dai.

Digital Object Identifier 10.1109/OJCS.2021.3100870

Joint Computation Offloading, Channel Access and Scheduling Optimization in UAV Swarms: A Game-Theoretic Learning Approach

RUNFENG CHEN [1], LI CUI [1], MENG WANG [1], YULI ZHANG [2], KAILING YAO [1], YANG YANG [1], AND CHANGHUA YAO [3]

[1]College of Communication Engineering, PLA Army Engineering University, Nanjing 210014, China
[2]National Innovation Institute of Defense Technology, Academy of Military Sciences, Beijing 100071, China
[3]School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China

CORRESPONDING AUTHOR: YULI ZHANG (e-mail: yulipkueecs08@126.com)

This work was supported in part by the National Science Foundation of China under Grants 61771488, 61901506, 61901520, and 61971439, in part by the Natural Science Foundation of Jiangsu Province of China under Grant BK20191329, in part by the Postdoctoral Science Fund of China under Grant 2019T120987, and in part by the Startup Foundation for Introducing Talent of NUIST under Grant 2020r100.
This paper has supplementary downloadable material available at http://ieeexplore.ieee.org.

ABSTRACT Coalition-based unmanned aerial vehicle (UAV) swarms have been widely used in urgent missions. To fasten the completion, mobile edge computing (MEC) has been introduced into UAV networks where coalition leaders act as servers to help members with data computing. This paper investigates a relative delay optimization in MEC-assisted UAV swarms. Considering that the scheduling methods have great impact on the delay, some theoretical analyses are made and a scheduling method based on the shortest effective job first (SEJF) is proposed. Based on the coupled relationship between scheduling and resource allocation, the computation offloading and channel access problems are then jointly optimized. To solve the problem in distributed UAV networks, the optimization problem is formulated as an offloading game. It is proved that the game is an exact potential game (EPG) and it has at least one pure strategy Nash Equilibrium (PNE). To reach the PNE, a distributed offloading algorithm based on concurrent best-better response (CBBR) is designed. Finally, the simulations show that the performance of the proposed CBBR algorithm is better than traditional algorithms. Compared with other scheduling methods, the proposed scheduling method based on SEJF reduces the delay by up to 30%.

INDEX TERMS UAV swarms, mobile edge computing, delay, scheduling, potential game.

I. INTRODUCTION

With the advantages of flexibility, intelligence and diversity, coalition-based unmanned aerial vehicle (UAV) swarms have been widely applied in urgent missions, e.g., search and rescue [1]–[4]. The completion time is crucial for these missions. As a promising technique, mobile edge computing (MEC) can effectively help UAV networks reduce the delay by offloading coalition members' computation to the leader [5]–[7]. In MEC-assisted UAV networks, how to schedule and utilize computing and spectrum resources directly affect the computing delay [8]–[10]. Most existing papers assumed the servers computed data for several users simultaneously [11]–[14], which only optimized the energy consumption or economic revenues. Besides, a few papers aimed to improve the performance of long term missions [15], [16], which could not be well applied in emergent UAV missions. Therefore, we jointly optimize computation offloading, channel access and scheduling in MEC-assisted UAV networks.

Due to the large-scale network and limited resources, there are still several challenges to solve the above optimization problem. Firstly, the MEC in coalition scenarios is more complicated than traditional networks (device-to-device networks and cellular networks). Within a UAV coalition, the members share the computing resource and the scheduling method should consider the impact of queuing on delay. Among UAV coalitions, coalitions share the spectrum resource and the interference should be considered [12], [17], [18]. Secondly, the resources allocation and computation offloading are coupled. On one hand, due to limitation of computing resource, offloading by many members increase the computing delay. On the other hand, the more data a member offloads, the longer the transmission delay [19]–[22]. Thirdly, it's difficult to realize the centralized decision-making due to the large-scale nature of UAV networks [23]–[26]. The UAV coalitions should make their decisions in a distributed way.

To tackle with the above problems, the relative delay optimization based on a game-theoretic learning approach is studied in this paper. Specially, we design a scheduling method based on shortest effective job first (SEJF) and build the offloading model, where UAV members choose both offloading strategy and transmission channels. Then, the optimization problem is formulated as a delay minimization game which is proved to be an exact potential game (EPG). To reach the pure strategy Nash Equilibrium (PNE), an offloading algorithm based on concurrent best-better response (CBBR) is proposed.

The main contributions of this paper are summarized as follows:

- To shorten the relative delay of UAV networks, the impact of scheduling methods on delay is analyzed and the scheduling method based on SEJF is proposed. Considering the limitation of computing and spectrum resource, the joint computation offloading and channel access optimization is investigated.
- In order to solve the optimization problem in distributed UAV coalitions, a game-theoretic learning method is utilized. The problem is first formulated as a game model and it is proved to be an EPG admitting at least one PNE. The offloading algorithm based on concurrent best-better response is designed to reach the PNE of the proposed game.
- Simulations show that the proposed CBBR algorithm converges fast. Besides, compared with other scheduling methods, the proposed scheduling method shortens the delay by up to 30%.

A preliminary version of this work was [27] and the extensions of this paper are concluded as follows: 1) The heterogeneous data in UAV networks is considered and the optimization objective is adjusted to relative delay. 2) The scheduling method of UAV leaders is newly designed in this paper. Compared with the first come first served (FCFS) method in [27], the proposed scheduling method based on SEJF saves delay. 3) Based on better response algorithm, the concurrent best-better response algorithm is proposed and has well convergence performance. 4) Some simulations are made to verify the proposed scheduling method and the proposed CBBR algorithm.

The rest of this paper is structured as follows. Related work is summarized in Section II. Section III builds the offloading model and formulates the optimization problem of computing delay in the UAV networks. Section IV formulates the optimization problem as a game model, analyzes the equilibrium and proposes a distributed offloading algorithm. Simulation results are presented and analyzed in Section V. Section VI summarizes the whole paper.

II. RELATED WORK

UAVs can carry out a variety of urgent and complex missions and many papers have studied UAV's applications [28]–[34]. Note that, in addition to performance on communications, UAV coalitions focus on the collaboration among UAVs and the performance of missions executions. As an effective technology to reduce the latency and energy consumption of mobile devices, MEC has been studied in many papers, which mainly optimized network delay [16], [18], [35], [36], energy consumption [17], [37], and the weighted sum of delay and energy consumption [20], [38], [39], [43]. Actually, UAVs' flight energy consumption is much larger than energy of transmission and computation. UAV networks carrying out urgent missions aim to complete missions as quickly as possible. Therefore, this paper considers the minimization of delay in coalition-based UAV networks.

Specially, [18] focused on latency minimization under the energy consumption and resource allocation constraints in a MEC-enabled device-to-device network, which considered partial offloading and interference. The author of [35] used the idea of software defined network and investigated the minimization of latency under the limitation of battery's life in ultra-dense network. To reduce the execution latency of devices with energy harvesting technologies in a recyclable system, [36] proposed a dynamic computing offloading approach. For UAVs scenario, [40] investigated the balance of UAVs' load and the minimization of task delay by UAVs' deployments. [41] studied a two-hop UAV model where a top UAV was considered as a MEC server, and the author optimized the response delay of the network. However, these papers mainly optimized the resources allocation, position of UAVs and offloading data's size. They didn't consider the impact of scheduling method of the computing resources on the delay.

When scheduling computing resources, [11]–[14] assumed that the servers provided computing service to several users simultaneously. These scheduling methods optimized the energy consumption or economic revenues, which couldnt cope with the minimization of delay. Besides, the scheduling methods in [15], [16] mainly aimed to long term tasks, of which the statistical characteristics could be observed. For urgent missions in UAV networks, it is hard to foresee their characteristics. Therefore, existing scheduling methods can't be applied to the UAV networks.

Many algorithms were proposed to solve the optimization problems in UAV-based MEC networks [8], [9], [40], [41]. They mainly used the convex optimization method, where either the UAV gathered the entire network information and made decision, or base stations helped to make decisions and sent the results to the users. However, in coalition-based

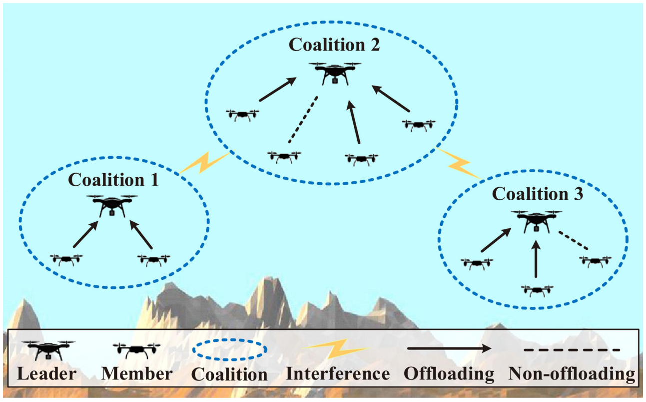

FIGURE 1. The demonstration of UAV coalitions executing missions.

UAV networks, each coalition makes decision by itself. Game-theoretic learning approach, as a powerful tool, can model the interactions among each member, analyze the equilibrium and solve the optimization problems in a distributed manner [31], [32]. In this paper, the offloading optimization is formulated as the game model and the distributed learning algorithm is designed.

III. SYSTEM MODEL AND PROBLEM FORMULATION

As shown in Fig. 1, multiple UAVs are divided into several coalitions according to mission requirements. Each coalition consists of a coalition leader and several coalition members. Generally, coalition leaders connect with commanders and other leaders to get the latest situation information. Besides, each leader has a well performance on computing, which is considered as a computing server. It is noted that each UAV member executes different missions with different hardware equipment. During the execution of the missions, each member first collects mission data, then processes the data and makes decision based on the computing result. The detailed process is shown below:

- Data collecting: All members first collect data. Due to the complexity of missions, UAV members have different division of labor and collect different types of data. For example, in a disaster relief scenario, some members are responsible for collecting information about people's vital signs, some members gather image data about environments, some members need to collect audio information and so on.
- Offloading: After data collecting, coalition members choose whether to offload data to the leader according to its own data size and real-time communication situation. If the member wants to offload data, it has to determine the transmission channels and the proportion of offloading data in the total data. Specially, in order to make full use of spectrum resource and reduce offloading delay, the members transmit data over multiple channels.
- Data computing: Assumed that the collected data is discontinuous, the computing is divided into local computing and remote computing. The offloading data is computed by the coalition leader, while the rest is computed locally by the members. The capability of data computing depends not only on the hardware conditions of the UAVs, but also on the software applications supported by the UAVs. To ensure the accuracy of the results, the leader won't start the computing until the data has been transferred [12].
- Result transmission: After the data has been computed by the leader, the result is transmitted from leader to the member. Because the size of the result is much less than the original data, the transfer time can be negligible.

A. OFFLOADING AND LOCAL COMPUTING

There are M coalitions in the network and the set of them is $\mathcal{M} = \{1, 2, ..., M\}$. In coalition m, the number of UAV members is N_m and the set of them is $\mathcal{N}_m = \{1, 2, ..., N_m\}$. The set of available channels is $\mathcal{A} = \{1, 2, ..., A\}$, from which members select several channels for data transmission. The set of data types is $\mathcal{X} = \{1, 2, ..., X\}$ and each member only collects one kind of data in subset $\mathcal{X}$.

For the coalition m, its leader's computing resource is f_m (cycles/s). Because computing different types of data requires different softwares, the leaders have different computing capabilities for heterogeneous data [42]. The computing efficiency for data x is defined as $\eta_{m,x}$ ($\eta \in [0, 1]$), which indicates the efficient matching capability of leader m to data x. For the ith member of coalition m, i.e., $n_{m,i}$, the data model is defined as $\langle X_{m,i}, L_{m,i}, C_{m,i}, \tau_{m,i} \rangle$, which is characterized by computing demands and specific features [43]. In the model, $X_{m,i}$ is the data type, $L_{m,i}$ (bits) is the size of total data, $C_{m,i}$ (cycles/bit) is the number of CPU cycles required to process 1-bit of data and reflects the computing demand, and $\tau_{m,i}$ (sec) is the time constraint which requires the completion of data computing.

The offloading strategy of $n_{m,i}$ is defined as $S_{m,i} = (\omega_{m,i}, \mathcal{K}_{m,i})$, where $\omega_{m,i} \in \Omega$ is the proportion of the offloading data to the total data and $\mathcal{K}_{m,i}$ is the channel selection of $n_{m,i}$. $\Omega = \{\omega_1, \omega_2, ..., \omega_{|\Omega|}\}$ is the set of available offloading ratio. If $n_{m,i}$ does not choose to offload the data, $\omega_{m,i} = 0$ and $\mathcal{K}_{m,i} = \{0\}$, otherwise $\omega_{m,i} \neq 0$ and $\mathcal{K}_{m,i} \subset \mathcal{A}$. The member won't offload all the data to the leader, otherwise it will waste the local computing resource and prolong the computing time. Therefore, $\omega_{m,i} \neq 1$. Considering the actual situation, the excessive number of selected channels not only has a higher requirement on the hardware, but may also affect the offloading efficiency of other users, so the maximum number of optional channels is 2. Hence, $\mathcal{K}_{m,i}$ is expressed as $\mathcal{K}_{m,i} = \{k^1_{m,i}\}$ or $\mathcal{K}_{m,i} = \{k^1_{m,i}, k^2_{m,i}\}$.

Due to the energy limitation of the UAVs, the power of the member remains constant when it transmits information. $P_{m,i}$ is denoted as the total transmission power of $n_{m,i}$ and

$|\mathcal{K}_{m,i}|$ is the number of its selected channels. The transmission power over single channel is $\frac{P_{m,i}}{|\mathcal{K}_{m,i}|}$. According to [44], the transmission data rate over the channel $k_{m,i}^j$ from $n_{m,i}$ to its leader is

$$R_{m,i}^j = B \cdot \log(1 + \frac{1}{|\mathcal{K}_{m,i}|} \cdot \frac{P_{m,i} d_{m,i}^{-\alpha}}{N_0 + \sum_{g\in \mathcal{J}_{m,i}} I_{g,j}}), \quad (1)$$

in which B is the bandwidth of one sub-channel, $d_{m,i}$ is the distance between $n_{m,i}$ and its leader, α is the pass loth, N_0 is the background noise, $\mathcal{J}_{m,i}$ is the set of its neighbor UAVs and $I_{g,j}$ is the interference of neighbor $\mathcal{J}_{m,i}(g)$ to $n_{m,i}$. In detail,

$$I_{g,j} = \delta(k_{m,i}^j \in \mathcal{K}_g) \cdot \frac{1}{|\mathcal{K}_g|} P_g D_{g,\mathcal{N}_m}, \quad (2)$$

where

$$\delta(\bullet) = \begin{cases} 1, & \bullet \text{ is true} \\ 0, & \bullet \text{ is false} \end{cases} \quad (3)$$

is indicator function, $\mathcal{K}_g$ is the channel selection of $\mathcal{J}_{m,i}(g)$, P_g is the power and $D_{g,\mathcal{N}_m} = d_{g,\mathcal{N}_m}^{-\alpha}$ is the channel gain from $\mathcal{J}_{m,i}(g)$ to $n_{m,i}$'s leader. The total transmission rate is mathematically expressed as

$$R_{m,i} = \sum_{k_{m,i}^j \in K_{m,i}} R_{m,i}^j. \quad (4)$$

Hence, the delay of data offloading is

$$T_{m,i}^{offload} = \frac{\omega_{m,i} L_{m,i}}{R_{m,i}}. \quad (5)$$

Assumed that the computing efficiency of each member for its own data is 100%. Therefore, the local computing delay of $n_{m,i}$ is

$$T_{m,i}^{local} = \frac{(1-\omega_{m,i}) L_{m,i} C_{m,i}}{f_{m,i}}, \quad (6)$$

where $f_{m,i}$(cycles/s) is the computing resource of $n_{m,i}$.

B. REMOTE COMPUTING

The offloading data is computed by the leader, the delay of remote computing of $n_{m,i}$ is

$$T_{m,i}^{comp} = \frac{\omega_{m,i} L_{m,i} C_{m,i}}{\eta_{m,i} f_m}, \quad (7)$$

where f_m is the computing resource of the leader in coalition m and $\eta_{m,i}$ is the computing efficiency of the leader to data of $n_{m,i}$.

In the process of remote computing, some offloading data need to queue in the cache queue of the leader. As shown in Fig. 2, $T_{m,i}^{oc}$ is defined as the delay from the start of offloading to the end of remote computing. The queuing delay is affected by the scheduling of computing resource.

The common scheduling methods include averaging resource, first come first served (FCFS) [45] and shortest job first (SJF) [46]. For heterogeneous data requirements in UAV networks, we propose a scheduling method based on shortest effective job first (SEJF). The detailed process is shown in Algorithm 1. When members' time constraints are different, the leader prioritizes the member with low time constraints. When members have the same time constraint, SEJF is applied in remote computing. To compare the performance of different scheduling method, some discussions are analyzed as follows:

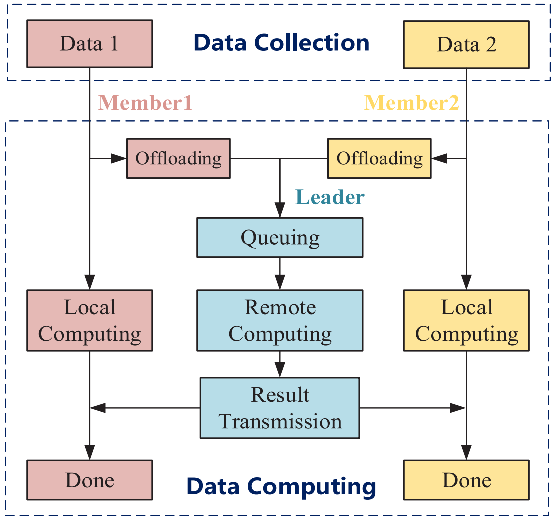

FIGURE 2. The processing diagram of data computing in UAV networks.

Algorithm 1: A Scheduling Method based on Shortest Effective Job First.

Step 1) If the computing resource is unused, the arrived data is computed immediately; Otherwise, execute Step 2.

Step 2) Calculate the effective job by (8) and add the data to the cache queue.

Step 3) When the current data has been computed, it's the turn of the data with the least time constraint to be computed. For the data with same time constraint, the data with shortest effective job is computed first.

Note that remote computing of one coalition is not affected by other coalitions. In one coalition, we assumed that the computing resource of leader is F and UAV member r is the rth member to finish offloading ($r = 1, 2, ..., N$), its offloading data's size is L_r. The effective job of member r is

$$W_r = \frac{L_r C_r}{\eta_r}. \quad (8)$$

In practice, the delay of remote computing is much longer than the offloading transmission delay, so it's assumed that the members in one coalition complete offloading almost simultaneously. According to the offloading order of UAV members, the effective job is $W_1, W_2, ..., W_N$. According to the order of workload from small to large, the effective job is $W^{(1)}, W^{(2)}, ..., W^{(N)}$. Then, the performance of scheduling methods is analyzed.

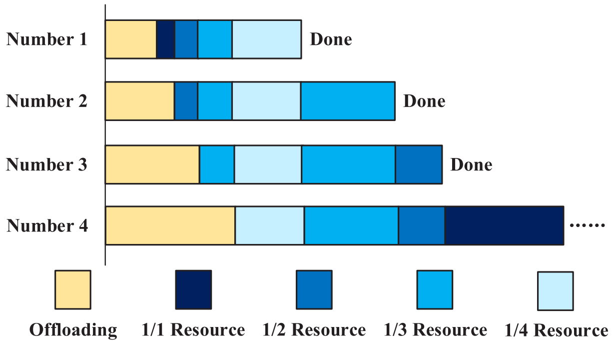

FIGURE 3. Illustration of averaging computing resource.

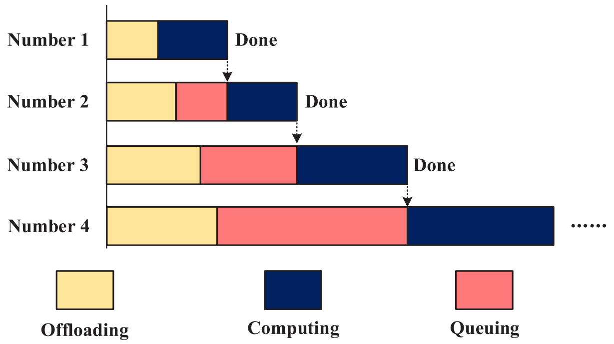

FIGURE 4. Illustration of first come first served.

1) AVERAGING COMPUTING RESOURCE

Averaging computing resource (ACR) means that the leader averages its computing resource to several parts depending on the amount of members whose offloading data has not been computed. For example, as shown in Fig. 3, when the first member finish offloading, it takes up all the computing resource. After the fourth member's data has been offloaded, each member takes up a quarter of the leader's resource.

Hence, for the nth member to complete remote computing, the delay from the end of offloading to the end of remote computing is

$$t_{ACR}(n) = \begin{cases} \frac{NW^{(1)}}{F}, & n = 1 \\ \frac{\sum_{r=1}^{n-1} W^{(r)} + (N-1)W^{(n)}}{F}, & n \geq 2 \end{cases}. \tag{9}$$

The total delay of remote computing is

$$T_{ACR}(N) = \frac{(2N-1)W^{(1)}}{F} + \sum_{r=2}^{N} \frac{(2N-r-1)W^{(r)}}{F}. \tag{10}$$

2) FIRST COME FIRST SERVED

Fig. 4 shows the model of first come first served in [27], where the members in a coalition are sorted according to arrival order. Different from averaging computing resource, UAV leaders only provide computing services for one UAV member at a time. The lower the order of member, the earlier its data start to be computed.

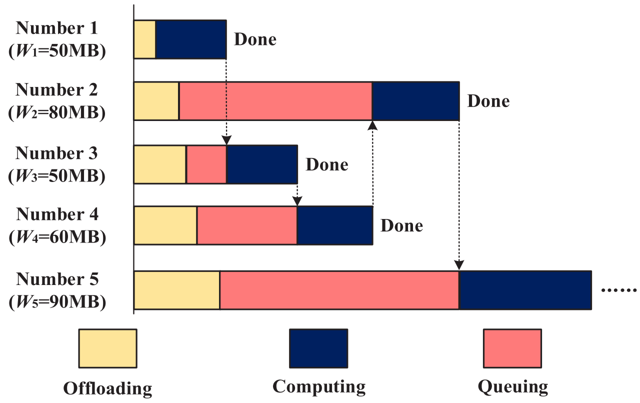

FIGURE 5. Illustration of shortest effective job first.

In the FCFS queuing model, the delay of the nth member to complete remote computing is

$$t_{FCFS}(n) = t_{FCFS}(n-1) + \frac{W_n}{F} = \sum_{r=1}^{n} \frac{W_r}{F}. \tag{11}$$

The total delay of remote computing under FCFS is

$$T_{FCFS}(N) = \sum_{n=1}^{N} \sum_{r=1}^{n} \frac{W_r}{F} = \frac{1}{F} \sum_{r=1}^{N} (N-r+1) W_r. \tag{12}$$

3) SHORTEST EFFECTIVE JOB FIRST

In the model of shortest job first, UAV leaders give priority to computing services for members with small data size. However, due to diversity of data types, SJF can't be applied directly in this scenario. Considering the data size and data type comprehensively, the SEJF scheduling mode is proposed. The detail is shown in Fig. 5.

In SEJF queuing model, the delay of the nth member to complete remote computing is

$$t_{SEJF}(n) = t_{SEJF}(n-1) + \frac{W^{(n)}}{F} = \sum_{r=1}^{n} \frac{W^{(n)}}{F}. \tag{13}$$

The total delay is

$$T_{SEJF}(N) = \sum_{n=1}^{N} \sum_{r=1}^{n} \frac{W^{(r)}}{F} = \frac{1}{F} \sum_{r=1}^{N} (N-r+1) W^{(r)}. \tag{14}$$

Theorem 1: The total delay of SEJF is no greater than that of ACR, i.e., $T_{SEJF} \leq T_{ACR}$.

Proof: For $N = 2$, $t_{ACR}(1) > t_{SEJF}(1)$ and $t_{ACR}(2) = t_{SEJF}(2)$, so $T_{ACR} > T_{SEJF}$. For $N > 2$, $t_{ACR}(n) > t_{SEJF}(n)$ is always true. Therefore, $T_{ACR} > T_{SEJF}$, i.e., SEJF is better than ACR. ■

Theorem 2: The total delay under SEJF is no greater than that of FCFS, i.e., $T_{SEJF} \leq T_{FCFS}$.

Proof: If the sequence $\{W_1, W_2, ..., W_N\}$ is identical with the sequence $\{W^{(1)}, W^{(2)}, ..., W^{(N)}\}$, $T_{FCFS} = T_{SEJF}$. When two sequences are different, the polynomial H is defined as

$$H = T_{FCFS}(N) = \frac{1}{F}(W_N + 2W_{N-1} + ... + NW_1). \quad (15)$$

For the case that W_N is not equal to $W^{(N)}$, W_k is found from sequence $\{W_1, W_2, ..., W_N\}$, which satisfies $W_k = W^{(N)}$. Then, W_k is swapped with $W^{(N)}$ in polynomial H and polynomial H_1 is formed:

$$H_1 = \frac{1}{F}(W_k + 2W_{N-1} + ... + kW_N + ... + NW_1), \quad (16)$$

$$\begin{aligned} H - H_1 &= \frac{1}{F}[W_N + (N+1-k)W_k] \\ &\quad - \frac{1}{F}[W_k + (N+1-k)W_N] \\ &= \frac{1}{F}(W_N - W_k)(-N+k) \geq 0. \end{aligned} \quad (17)$$

For the case that W_N is equal to $W^{(N)}$, the element of which value is equal to $W^{(N-1)}$ will be swapped. And so on, polynomial H_1 is adjusted to form polynomial H_2. It's easy found that H_1 is larger than H_2. After a finite number of exchanges,

$$H' = \frac{1}{F}(L^{(N)} + 2L^{(N-1)} + ... + NL^{(1)}) = T_{SEJF}(N), \quad (18)$$

it is seen that $H > H'$, i.e., $T_{FCFS}(N) > T_{SEJF}(N)$, SEJF is better than FCFS. ■

To sum up, shortest effective job first is the best way to schedule computing resource among these three methods when members have the same time constraint. Therefore, UAV coalitions in proposed system model adopt the scheduling method based on SEJF.

C. PROBLEM FORMULATION

Since the delay of result transmission is negligible, the total delay for $n_{m,i}$ to complete the data computing is

$$T_{m,i} = \max[T_{m,i}^{local}, T_{m,i}^{oc}]. \quad (19)$$

Relative delay is defined as the quotient of the absolute delay divided by the time constraint, which reflects the computational efficiency of the data, i.e.,

$$RT_{m,i} = \frac{T_{m,i}}{\tau_{m,i}}. \quad (20)$$

Because missions data among members are not relevant, the relative delay of the network is defined as the average of all members' delay,

$$RT = \frac{1}{M}\sum_{m\in\mathcal{M}}\left(\frac{1}{N_m}\sum_{i\in\mathcal{N}_m} RT_{m,i}\right). \quad (21)$$

For members with low time constraints, the mission should be completed as quickly as possible. Therefore, the optimization for relative delay is more significant than that for absolute delay. The strategy space of all coalition members is $\mathcal{S} = \Omega \otimes \mathcal{A}'$, where $\mathcal{A}' = \mathcal{A} \cup \{0\} \cup \{k_1, k_2 | k_1 \neq k_2, k_1 \in \mathcal{A}, k_2 \in \mathcal{A}\}$ is the strategy space of channel selections. The strategy of the network is defined as $S = \{S_{m,i} | m = 1, 2, ..., M; i = 1, 2, ..., N_m\}$, where $\forall m \in \mathcal{M}, \forall i \in \mathcal{N}_m, S_{m,i} \in \mathcal{S}$.

Accordingly, the delay optimization problem is formulated as

$$(\text{OP}) : \min_{S} RT, \quad (22)$$

s.t.

$$\begin{aligned} &\mathcal{K}_{m,i1} \cap \mathcal{K}_{m,i2} = \emptyset, \forall \mathcal{K}_{m,i1}, \mathcal{K}_{m,i2} \in \mathcal{A}', \\ &i1, i2 \in \mathcal{N}_m, m \in \mathcal{M}. \end{aligned} \quad (23)$$

IV. GAME ANALYSIS AND LEARNING ALGORITHM

A. GAME MODEL

The proposed optimization problem is a joint optimization problem in decentralized scenario, in which there exists a huge strategy space. Game theory has been widely used in resource optimizations of distributed wireless networks due to its low computational complexity and well effectiveness [47]–[52]. Hence, the offloading game model is formulated as

$$G = [\mathcal{M}, \mathcal{N}, \{U_{m,i}\}_{m\in\mathcal{M}, i\in\mathcal{N}_m}, \{S_{m,i}\}_{m\in\mathcal{M}, i\in\mathcal{N}_m}], \quad (24)$$

in which $\mathcal{M}$ is the set of coalitions, $\mathcal{N}$ is the set of UAV members and $U_{m,i}$ is the utility function of $n_{m,i}$. It can be found that when the member changes its own strategy, it has an impact on other members in same coalition, the neighbors and neighbors' coalition members. Motivated by [49], the utility function of $n_{m,i}$ is designed as

$$U_{m,i}(S_{m,i}, S_{-m,i}) = -\left(\sum_{n\in\mathcal{N}_m} RT_n + \sum_{g\in\mathcal{J}_{m,i}}\sum_{j\in\mathcal{N}_g} RT_{g,j}\right), \quad (25)$$

where $S_{m,i}$ is the offloading strategy of $n_{m,i}$, while $S_{-m,i}$ denotes the offloading strategies of all members in the network except $n_{m,i}$, $\mathcal{N}_m$ is the set of members which are in the same coalition with $n_{m,i}$, $\mathcal{J}_{m,i}$ is the set of coalitions which include $n_{m,i}$'s neighbors. Each UAV member maximizes its utility function as follows:

$$\max_{S_{m,i}\in\mathcal{S}} U_{m,i}(S_{m,i}, S_{-m,i}), \forall i \in \mathcal{N}_m, \forall m \in \mathcal{M}. \quad (26)$$

Definition 1: (Pure Nash Equilibrium (PNE) [33]): A strategy profile $S^{*\rightarrow} = (S_1^*, S_2^*, ..., S_N^*)$ is PNE of G if and only if no member can gain more profit by deviating its strategy. Mathematically,

$$\begin{aligned} &U_{m,i}(S_{m,i}^*, S_{-m,i}^*) \geq U_{m,i}(S_{m,i}, S_{-m,i}^*), \\ &\forall m \in \mathcal{M}, i \in \mathcal{N}_m, S_{m,i} \in \mathcal{S}, S_{m,i} \neq S_{m,i}^*. \end{aligned} \quad (27)$$

Definition 2: (Exact Potential Game (EPG) [34]): A game G is an EPG if there exists a potential function, which satisfies

$$\begin{aligned} \phi(\overline{S_{m,i}}, S_{-m,i}) - \phi(S_{m,i}, S_{-m,i}) &= U_{m,i}(\overline{S_{m,i}}, S_{-m,i}) \\ &\quad - U_{m,i}(S_{m,i}, S_{-m,i}), \end{aligned}$$

$$\forall m \in \mathcal{M}, i \in \mathcal{N}_m, S_{m,i} \in \mathcal{S}, \overline{S_{m,i}} \in \mathcal{S}, S_{m,i} \neq \overline{S_{m,i}}. \tag{28}$$

For an EPG, it has at least one pure Nash Equilibrium and the best PNE maximizes the potential function [49].

B. GAME EQUILIBRIUM ANALYSIS

Theorem 3: The proposed offloading game G is an EPG.

Proof: Firstly, the potential function is constructed as

$$\phi(S_{m,i}, S_{-m,i}) = -\sum_{p \in \mathcal{M}} \sum_{q \in \mathcal{N}_p} RT_{p,q}, \tag{29}$$

which is the negative value of total delay of all members in the network.

If an arbitrary member $n_{m,i}$ unilaterally changes its offloading action from $S_{m,i}$ to $\overline{S_{m,i}}$, there are mainly three cases: only changing the offloading ratio, only changing the channel access strategy and changing both.

- The member always chooses to offload and only change the offloading ratio. In this case, only the mission delay of the member and members in same coalition will be affected,

$$\begin{aligned}
&\phi(\overline{S_{m,i}}, S_{-m,i}) - \phi(S_{m,i}, S_{-m,i}) \\
&\quad = \sum_{p \in \mathcal{M}} \sum_{q \in \mathcal{N}_p} RT_{p,q}(\overline{S_{m,i}}, S_{-m,i}) \\
&\qquad - \sum_{p \in \mathcal{M}} \sum_{q \in \mathcal{N}_p} RT_{p,q}(S_{m,i}, S_{-m,i}) \\
&\quad = \sum_{q \in \mathcal{N}_m} RT_{m,q}(\overline{S_{m,i}}, S_{-m,i}) - \sum_{q \in \mathcal{N}_m} RT_{m,q}(S_{m,i}, S_{-m,i}) \\
&\quad = U_{m,i}(\overline{S_{m,i}}, S_{-m,i}) - U_{m,i}(S_{m,i}, S_{-m,i}).
\end{aligned} \tag{30}$$

- When only the channel access strategy is changed, or the offloading ratio and channel strategy are both changed, other members in same coalition, the neighbors and neighbors' coalition members will be influenced,

$$\begin{aligned}
&\phi(\overline{S_{m,i}}, S_{-m,i}) - \phi(S_{m,i}, S_{-m,i}) \\
&\quad = \sum_{p \in \mathcal{M}} \sum_{q \in \mathcal{N}_p} RT_{p,q}(S_{m,i}, S_{-m,i}) \\
&\qquad - \sum_{p \in \mathcal{M}} \sum_{q \in \mathcal{N}_p} RT_{p,q}(\overline{S_{m,i}}, S_{-m,i}) \\
&\quad = \sum_{g \in \mathcal{J}_{m,i}} \sum_{j \in \mathcal{N}_g} RT_{g,j}(S_{m,i}, S_{-m,i}) \\
&\qquad - \sum_{g \in \mathcal{J}_{m,i}} \sum_{j \in \mathcal{N}_g} RT_{g,j}(\overline{S_{m,i}}, S_{-m,i}) \\
&\qquad + \sum_{q \in \mathcal{N}_m} RT_{m,q}(S_{m,i}, S_{-m,i}) \\
&\qquad - \sum_{q \in \mathcal{N}_m} RT_{m,q}(\overline{S_{m,i}}, S_{-m,i}) \\
&\quad = U_{m,i}(\overline{S_{m,i}}, S_{-m,i}) - U_{m,i}(S_{m,i}, S_{-m,i}).
\end{aligned} \tag{31}$$

To sum up, when the UAV member unilaterally changes its offloading strategy, the change of potential function is equal to the change of its utility function, i.e.,

$$\begin{aligned}
\phi(\overline{S_{m,i}}, S_{-m,i}) - \phi(S_{m,i}, S_{-m,i}) &= U_{m,i}(\overline{S_{m,i}}, S_{-m,i}) \\
&\quad - U_{m,i}(S_{m,i}, S_{-m,i}).
\end{aligned} \tag{32}$$

■

Therefore, the best PNE of the offloading game G corresponds to the optimal solution of formula (22). As each member in the network maximizes utility function respectively by changing offloading strategies, the relative delay will reach the minimum.

C. DISTRIBUTED LEARNING ALGORITHM

When solving the optimal strategy in distributed scenario, the available algorithms include best response [47], better response [50], stochastic learning automata [51] and spatial adaptive play [34]. In best response algorithm, the best strategy for a certain member is selected in each iteration, which causes the algorithm to have high spatial complexity. While the better response does not cost more computational memory, it needs more iteration times to reach the convergence. Compromising between space complexity and time complexity, best-better response is proposed. Motivated by [33], concurrent algorithms could further improve the convergence speed. Therefore, an offloading algorithm based on concurrent best-better response is designed. The specific process is shown as Algorithm 2.

It is assumed that each UAV leader has two transceivers which operate on different sub-channels. One transceiver is used to communicate with leaders in a common control channel while the other is used for sensing and intra-coalition communication. In order to select non-neighbor members, motivated by [33], [49], 802.11 DCF-like contention mechanism is applied. The detailed steps are as follows:

1) Coalition leaders generate a backoff timer according to uniform distribution in $[0, \tau_{\max}]$ for fixed parameter $\tau_{\max}$.

2) When the backoff timer expires, the leader observes the common control channel and send an updating request-to-send packet in the channel to update its member's offloading strategy.

3) Once hearing the updating message by others, all neighbor leaders freeze their backoff timer and keep silent until the next iteration.

The leaders execute algorithms by information interaction. After the algorithm converges, leaders sent the optimal strategies to their members. Especially, best response is applied to select the best offloading ratio and the better response is used to select the channels.

Theorem 4: The CBBR algorithm converges to the PNE of offloading game model G, which is the optimal solution locally or globally of the proposed optimization problem.

Algorithm 2: Offloading Algorithm Based on Concurrent Best-Better Response.

Initialize: UAV members in the network randomly select the offloading ratio and transmission channels from the strategy space.

Loop:

Step 1) All UAV members interact with their coalition leaders;

Step 2) Select multiple non-neighborhood UAV members randomly for updating strategies. For any two selected members $n_{m1,i1}$ and $n_{m2,i2}$, $m1 \notin \mathcal{J}_{m2,i2}$ and $m2 \notin \mathcal{J}_{m1,i1}$. The specific process is as follows:

Step 3) For the selected members in $\mathcal{N}_s$, each member first selects the offloading ratio that maximizes its utility function respectively,

$$\omega_{m,i} = \arg \max_{\omega_{m,i} \in \Omega} U_{m,i}(\omega_{m,i}, \mathcal{K}_{m,i}, S_{-m,i}). \tag{33}$$

Then, the members update strategies of channels. If the new channel strategy satisfies

$$U_{m,i}(\omega_{m,i}, \mathcal{K}^*_{m,i}, S_{-m,i}) \geq U_{m,i}(\omega_{m,i}, \mathcal{K}_{m,i}, S_{-m,i}), \tag{34}$$

$\mathcal{K}^*_{m,i} = \mathcal{K}_{m,i}$, otherwise $\mathcal{K}_{m,i}$ remains the same;

End Loop: Until the maximum number of iterations is reached or the state of convergence is entered.

Proof: For best response algorithm, the members updating the strategy in each iteration only select the strategy corresponding to the value of the maximum utility function. For better response algorithm, the member selects the strategy which makes the utility function nondecreasing. Therefore, in each iteration in CBBR, the utility function of the member either increases or remains at the currently optional maximum. This means that the potential function increases or remains constant in each iteration. At the same time, the potential function has its upper limit, so the proposed algorithm must converge to one of the NE after finite iterations. This completes the proof. ■

Both best response and better response, select one user randomly for iterative updating, and the convergence state is achieved after multiple iterations. However, if the number of users is large, the number of iterations required for the convergence will be very large, which seriously affects the efficiency of algorithm. The proposed CBBR algorithm selects multiple members in one iteration. These members are independent of each other and their own decision does not affect the utility of other members, which greatly improves the convergence speed. Since the nature of the CBBR is still the better response algorithm, the optimality of the algorithm solution will not be influenced. Therefore, the proposed algorithm improves the convergence speed while ensuring the optimal solution.

To analyze the complexity of the proposed offloading algorithm based on CBBR and its feasibility, we give the computational complexity and storage size as shown in Table I [33].

TABLE I. Summation of Used Notations

Computation	Operation	Storage Size	Complexity
—	member selection	—	$O(C_1)$
$U_{m,i}(\omega_{m,i})$	—	—	$O(C_2)$
$\omega_{m,i}$	comparison	$\|\Omega\|$	$O(C_3)$
$U_{m,i}(\mathcal{K}_{m,i})$	—	—	$O(C_4)$
$\mathcal{K}_{m,i}$	comparison	2	$O(C_5)$

TABLE II. Complexity Analysis of the Proposed CBBR Algorithm

Computation	Operation	Storage Size	Complexity
—	member selection	—	$O(C_1)$
$U_{m,i}(\omega_{m,i})$	—	—	$O(C_2)$
$\omega_{m,i}$	comparison	$\|\Omega\|$	$O(C_3)$
$U_{m,i}(\mathcal{K}_{m,i})$	—	—	$O(C_4)$
$\mathcal{K}_{m,i}$	comparison	2	$O(C_5)$

$\mathcal{N}_{it}$ is denoted as the number of iterations for the convergence. The detailed analysis process is shown as follows:

- **Non-neighbor Members:** In Step 2, non-neighbor members are selected by the leaders and the complexity is $O(C_1)$, where C_1 is a small constant influenced by the network scale.
- **Offloading ratio:** The best response algorithm is applied when members select the best offloading ratio. The members calculate the utility function and the complexity is $O(C_2)$, where C_2 is a small constant decided by equation (21). Then, the members find the best ratio by comparisons and the complexity is $O(C_3)$, where C_3 is influenced by the number of elements in the set Ω.
- **Access Channel:** The better response algorithm is used to select the access channel by members. The complexity of calculation is $O(C_4)$, where C_4 is a small constant. Members compare two utility function values and the complexity is expressed as $O(C_5)$.

Therefore, the computation complexity of the offloading algorithm based on CBBR is

$$O = \mathcal{N}_{it}[O(C_1) + O(C_2) + O(C_3) + O(C_4) + O(C_5)]. \tag{35}$$

It is noted that the complexity is mainly decided by the convergence iterations $\mathcal{N}_{it}$, which is influenced by the number of UAVs in the network. Besides, the memory space of UAVs is limited. In the proposed algorithm, the storage size is the number of utilities, i.e., $|\Omega|$ and 2, respectively. In a word, the proposed algorithm has low complexity, and the method can be implemented using typical UAV networks, which is indicated in the following simulation part.

V. SIMULATION SETUP AND EXPERIMENT RESULTS

A. SIMULATION SETUP

A 10000m × 10000m square is constructed, which consists of 16 mission regions. As shown in Fig. 6, a UAV coalition with 1 leader and 5 members, is randomly distributed in mission regions and each region contains at most one coalition. There

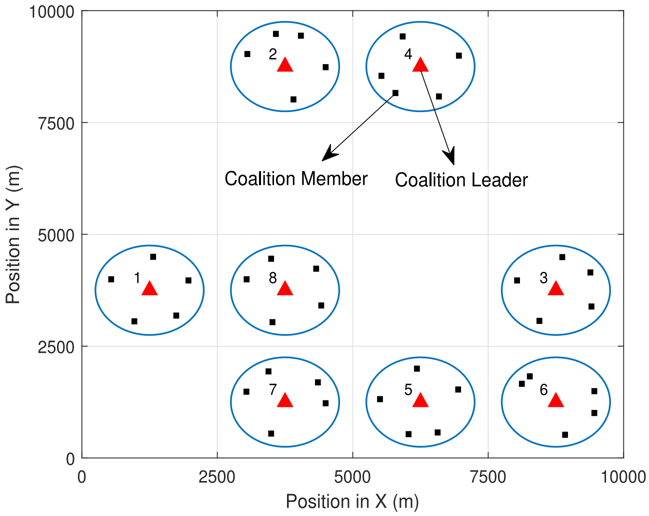

FIGURE 6. **A simulation network with eight coalitions.**

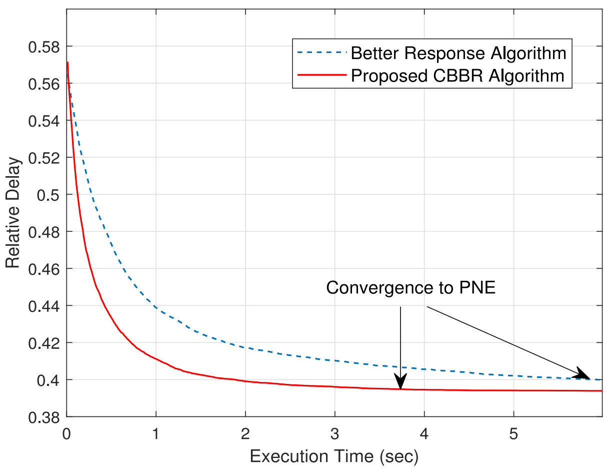

FIGURE 7. **Comparison of execution time under different algorithms.**

are 10 available channels with bandwidth $B = 5$MHz. The noise power is $N_0 = -100$dBm, the path loss factor is $\alpha = 5$, and the power of members is 0.02W. The computing resource of leaders and members are 12GHz and 3GHz respectively. The data size and the number of CPU cycles required to process 1-bit of data obey the uniform distribution in [20, 100] MB and [100,500] cycles. The time constraint is set as $4s$. Any two members from different coalitions are neighbors if their distance is less than 3000 m. The set of available offloading ratio is $\Omega = \{0, 0.1, 0.2, ..., 0.9\}$. Other parameters are given in the following specific scenarios. Note that, 500 experiments are conducted for each scenario respectively and the value of delay is the average result.

B. CONVERGENCE & PERFORMANCE

Fig. 7 shows convergence of different algorithms in the network with 4 coalitions. After a period of execution, both the better response (BR) and the CBBR algorithm reach the convergence. Compared with BR algorithm, the relative delay

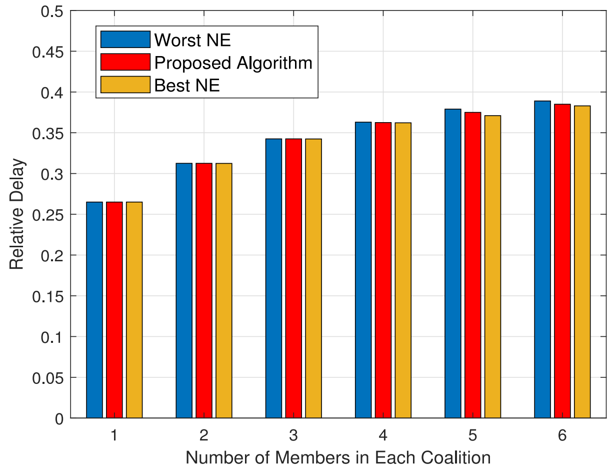

FIGURE 8. **Comparison of the performance under worst NE, best NE and the proposed algorithm.**

of the proposed CBBR algorithm is decreased by 3%. Because multiple members update strategies simultaneously in each iteration, the proposed algorithm needs less execution time to reach the convergence state. The proposed CBBR algorithm spends 3.7sec and saves almost 40% of execution time compared with BR.

The network and the space of strategies is so large that it is hard to solve the optimal solutions by exhaustive search. To analyze the optimality of the PNE, Fig. 8 compares the utility performance of three approaches, i.e., worst NE, best NE and proposed algorithm. The worst NE and best NE are the worst solution and the best solution respectively among 500 experiments and the approach of proposed algorithm is the average of all NEs. They are regarded as the lower bound and the upper bound of the proposed algorithm, where the best NE could be approximately equivalent to the optimal solution. It is observed that the solution of proposed algorithm is very closed to the best NE. Besides, the gap between the worst NE and the solution of proposed algorithm is very small, which indicates that the algorithm has a well performance.

C. SCALABILITY

To verify the scalability of the proposed algorithm, Fig. 9 shows the execution time for convergence to PNE, both for an increase number of coalitions and an increase number of members in each coalition [43]. From the figure, the execution time tends to increase linearly as the number of members in each coalition increases, which indicates that the complexity of the algorithm is influenced by the scale of UAVs network. It is observed that the number of coalitions has a greater effect on execution time than the number of members.

In addition, the relationship between the relative delay and the scale of the network is shown in Fig. 10. Different from the execution time, the relative delay is mainly influenced by the number of members. The specific analysis is given later. To summarize, the proposed framework scales well as the the network gets larger.

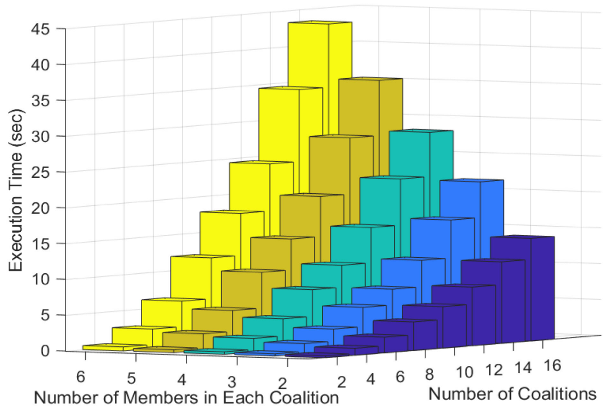

FIGURE 9. The relationship between execution time and scale of network.

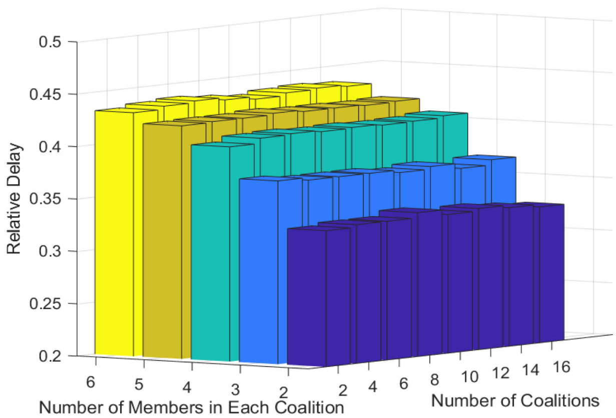

FIGURE 10. The relationship between relative delay and scale of network.

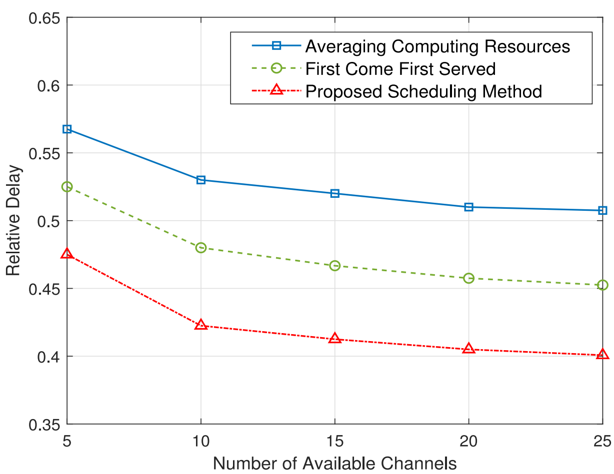

FIGURE 11. The relationship between relative delay and the number of available channels.

D. INFLUENCE OF CHANNEL RESOURCE

Fig. 11 shows the relationship between channel resource and relative delay in different scheduling modes. In the scenario, there are 8 coalitions and each coalition has 5 members. As the number of available channels increases, the delay decreases gradually. Note that, when the number goes from 5 to 10, the descend of the delay is more obvious. The reason is that each member only uses one channel when there are 5 available channels in the network. When the number of channels is 10, members transfer offloading data on double channels. However, although the number of channels increases, due to limitation of computing resource, the relative delay finally tends to stabilize.

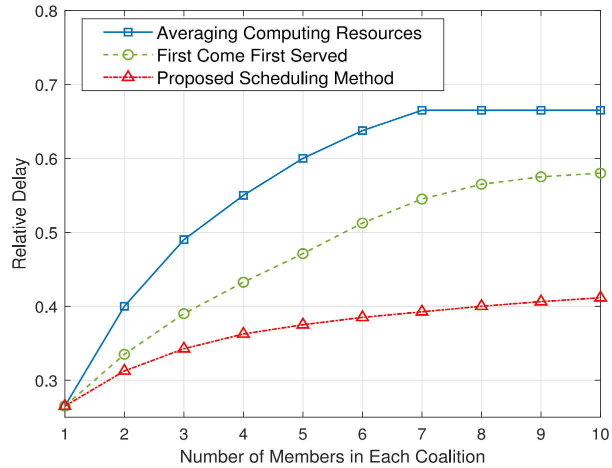

FIGURE 12. The relationship between relative delay and the number of coalition members.

E. INFLUENCE OF NUMBER OF MEMBERS

Fig. 12 shows the influence of members number on relative delay. There are 4 coalitions in the scenario. As the number of members in a coalition increases, the relative delay also increases. One hand, the computing delay is affected by the number of UAV members because of the limitation of computing resource. On the other hand, the increase in the number of members results in more interference, which leads to the increase of the transmission delay.

For different methods of scheduling, the proposed scheduling method based on SEJF brings in the lowest delay and the delay of ACR is the highest. Note that, the rate of change of the delay is getting slower. Especially, when the number is more than 7, the delay of ACR remains constant. The reason is that in the case, few members would like to offload data to the leader due to long queuing delay and limited computing resource.

F. INFLUENCE OF DATA SIZE

Fig. 13 shows the relationship with data size and the average delay in different scheduling methods. It is noted that the "data size" is the mean of all members' data size. No matter which method is adopted, the relationship between delay and data size is linear. This indicates that the offloading strategies remain same roughly when only the data size changes. Besides, as data size increases, the advantage of proposed scheduling method is more and more obvious. In this scenario, the scheduling method improves performance by 30% and 16% compared with ACR and FCFS respectively. Note that,

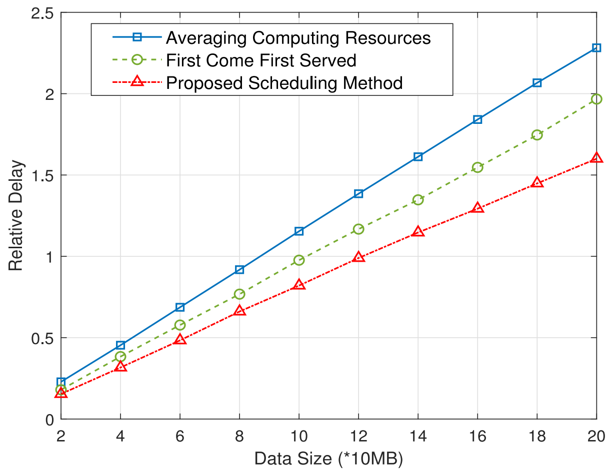

FIGURE 13. The relationship between relative delay and data size.

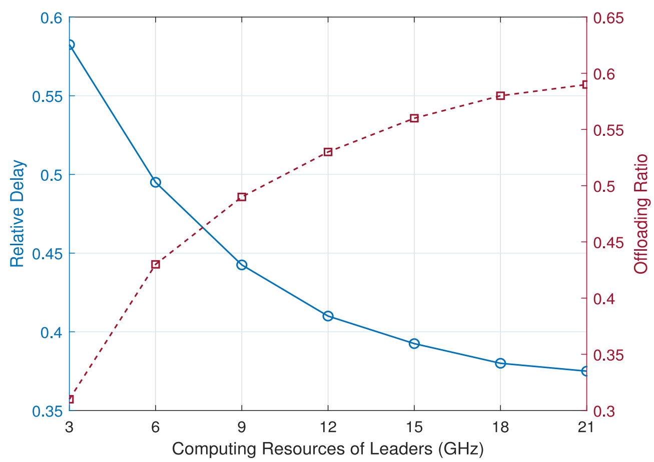

FIGURE 14. Relative delay and offloading ratio under different computing resource of leaders.

in a time constraint, the amount of data that UAV members compute is limited.

G. INFLUENCE OF COMPUTING RESOURCE

As shown in the Fig. 14, the blue curve depicts the relationship between computing resource of leaders and relative delay. The red one shows the change of average offloading ratio. When there are more computing resources, the delay is lower. It is noted that the rate of descent of the delay gradually decreases. This's because that the influence of computing resource is the main determinant in the earlier stage. The transmission delay and queuing delay limit the decrease of total delay in the later stage. The offloading ratio has an upper bound where the member's remote delay is approximately equal to local delay. If the member's offloading ratio exceeds the bound, the remote delay increases while the local delay decreases, which makes the delay longer. Therefore, commanders should reasonably configure the computing resource of UAVs according to the performance limit and economic conditions.

VI. CONCLUSION

In this paper, to minimize the total delay for heterogeneous data in MEC-assisted UAV networks, a joint computing offloading and channel access optimization based on SEJF scheduling was investigated, where UAVs adjusted the offloading ratio and channel access strategies comprehensively. A scheduling method based on shortest effective job first scheduling was designed to shorten the delay. To solve the problem in the distributed scenario, the offloading optimization problem was formulated as a game model, which was proved to be an EPG admitting at least one PNE. An offloading algorithm based on concurrent best-better response was adopted to reach the PNE. The simulation results strictly verified the reliability of the model and the validity of the proposed scheduling method. Hence, this study has a great application prospect in the UAV network based on coalitions.

REFERENCES

[1] Y. Zeng, R. Zhang, and T. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 54, no. 11, pp. 36–42, May 2016.
[2] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Commun. Surv. Tuts.*, vol. 18, no. 4, pp. 2624–2661, Oct.–Dec. 2016.
[3] D. Liu *et al.*, "Task-driven relay assignment in distributed UAV communication networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 11003–11017, Nov. 2019.
[4] D. Liu *et al.*, "Opportunistic UAV utilization in wireless networks: Motivations, applications, and challenges," *IEEE Commun. Mag.*, vol. 58, no. 5, pp. 62–68, May 2020.
[5] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surv. Tuts.*, vol. 19, no. 4, pp. 2322–2358, Oct.–Dec. 2017.
[6] 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.
[7] P. Mach and Z. Becvar, "Mobile edge computing: A. survey on architecture and computation offloading," *IEEE Commun. Surv. Tuts.*, vol. 19, no. 3, pp. 1628–1656, Jul.–Sep. 2017.
[8] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, "Computation rate maximization in UAV-enabled wireless-powered mobile-edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1927–1941, Sep. 2018.
[9] Q. Hu, Y. Cai, G. Yu, Z. Qin, M. Zhao, and G. Y. Li, "Joint offloading and trajectory design for UAV-enabled mobile edge computing systems," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1879–1892, Apr. 2019.
[10] 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.*, Kansas City, MO, USA, 2018, pp. 1–6.
[11] X. Hu, K. Wong, K. Yang, and Z. Zheng, "UAV-Assisted relaying and edge computing: Scheduling and trajectory optimization," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4738–4752, Oct. 2019.
[12] X. Chen, L. Jiao, W. Li *et al.*, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE Trans. Netw*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.
[13] W. Sun, J. Liu, Y. Yue, and H. Zhang, "Double auction-based resource allocation for mobile edge computing in Industrial Internet of Things," *IEEE Trans. Ind. Inform.*, vol. 14, no. 10, pp. 4692–4701, Oct. 2018.
[14] H. Zhang, F. Guo, H. Ji, and C. Zhu, "Combinational auction-based service provider selection in mobile edge computing networks," *IEEE Access*, vol. 5, pp. 13455–13464, 2017.
[15] L. Lei, H. Xu, X. Xiong, K. Zheng, and W. Xiang, "Joint computation offloading and multiuser scheduling using approximate dynamic programming in NB-IoT edge computing system," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5345–5362, Jun. 2019.
[16] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Proc. IEEE Int. Symp. Inf. Theory*, Barcelona, Spain, 2016, pp. 1451–1455, 2016.

[17] H. Chen, D. Zhao, Q. Chen, and R. Chai, "Joint computation offloading and radio resource allocations in small-cell wireless cellular networks," *IEEE Trans. Green Commun. Netw.*, vol. 4, no. 3, pp. 745–758, Sep. 2020.
[18] U. Saleem, Y. Liu, S. Jangsher, X. Tao, and Y. Li, "Latency minimization for D2D-enabled partial computation offloading in mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4472–4486, Apr. 2020.
[19] A. Qin, C. Cai, Q. Wang, Y. Ni, and H. Zhu, "Game theoretical multi-user computation offloading for mobile-edge cloud computing," in *Proc. IEEE Conf. Multimedia Inf. Process. Retrieval*, San Jose, CA, USA, 2019, pp. 328–332.
[20] R. Chai, J. Lin, M. Chen, and Q. Chen, "Task execution cost minimization-based joint computation offloading and resource allocation for cellular D2D MEC systems," *IEEE Syst. J.*, vol. 13, no. 4, pp. 4110–4121, Dec. 2019.
[21] F. Guo, H. Zhang, H. Ji, X. Li, and V. C. M. Leung, "An efficient computation offloading management scheme in the densely deployed small cell networks with mobile edge computing," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2651–2664, Dec. 2018.
[22] N. Qi, N. I. Miridakis, M. Xiao, T. A. Tsiftsis, R. Yao, and S. Jin, "Traffic-aware two-stage queueing communication networks: Queue analysis and energy saving," *IEEE Trans Commun.*, vol. 58, no. 8, pp. 4919–4932, Aug. 2020.
[23] Z. Yu, Y. Gong, S. Gong, and Y. Guo, "Joint task offloading and resource allocation in UAV-enabled mobile edge computing," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3147–3159, Apr. 2020.
[24] Z. Zhu, L. P. Qian, J. Shen, L. Huang, and Y. Wu, "Joint optimisation of UAV grouping and energy consumption in MEC-enabled UAV communication networks," *IET Commun.*, vol. 14, no. 16, pp. 2723–2730, 2020.
[25] L. Ruan *et al.*, "Energy-efficient Multi-UAV coverage deployment in UAV networks: A game-theoretic framework," *China Commun.*, vol. 15, no. 10, pp. 194–209, 2018.
[26] N. Qi, M. Xiao, T. A. Tsiftsis, R. Yao, and S. Mumtaz, "Energy efficient two-tier network-coded relaying systems considering processing energy costs," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 999–1003, Jan. 2019.
[27] R. Chen *et al.*, "Delay optimization with FCFS queuing model in mobile edge computing-assisted UAV swarms: A game-theoretic learning approach," in *Proc. IEEE Int. Conf. Wireless Commun. Signal Process.*, 2020, pp. 245–250.
[28] J. Li *et al.*, "Joint optimization on trajectory, altitude, velocity, and link scheduling for minimum mission time in UAV-aided data collection," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1464–1475, Feb. 2020.
[29] X. Yu, J. Yao, L. Ruan, K. Yao, D. Liu, R. Chen, "3D deployment of Multi-UAV for energy-saving: A game-based learning approach," in *Proc. IEEE 5th Int. Conf. Comput. Commun.*, Chengdu, China, 2019, pp. 1332–1337.
[30] D. Liu *et al.*, "Opportunistic data collection in cognitive wireless sensor networks: Air CGround collaborative online planning," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8837–8851, Sep. 2020.
[31] X. Wang *et al.*, "Machine learning empowered spectrum sharing in intelligent unmanned swarm communication systems: Challenges, requirements and solutions," *IEEE Access*, vol. 8, pp. 89839–89849, 2020.
[32] W. Zafar, B. M. Khan, "A reliable, delay bounded and less complex communication protocol for multicluster FANETs," *Digit. Commun. Netw*., vol. 3, no. 1, pp. 30–38, 2016.
[33] J. Chen, Y. Xu, Q. Wu, Y. Zhang, X. Chen, and N. Qi, "Interference-aware online distributed channel selection for multicluster FANET: A potential game approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3792–3804, Apr. 2019.
[34] K. Yao *et al.*, "Self-organizing slot access for neighboring cooperation in UAV swarms," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2800–2812, Apr. 2020.
[35] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 587–597, Mar. 2018.
[36] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.
[37] M. Kamoun, W. Labidi, and M. Sarkiss, "Joint resource allocation and offloading strategies in cloud enabled cellular networks," in *Proc. IEEE Int. Conf. Commun.*, London, U.K., 2015, pp. 5529–5534.
[38] X. Diao, J. Zheng, Y. Wu, and Y. Cai, "Joint computing resource, power, and channel allocations for D2D-assisted and NOMA-based mobile edge computing," *IEEE Access*, vol. 7, pp. 9243–9257, 2019.
[39] Y. He, J. Ren, G. Yu, and Y. Cai, "Joint computation offloading and resource allocation in D2D enabled MEC networks," in *Proc. IEEE Int. Conf. Commun.*, Shanghai, China, 2019, pp. 1–6.
[40] L. Yang, H. Yao, J. Wang, C. Jiang, A. Benslimane, and Y. Liu, "Multi-UAV-enabled load-balance mobile-edge computing for IoT networks," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6898–6908, Aug. 2020.
[41] Q. Zhang, J. Chen, L. Ji, Z. Feng, Z. Han, and Z. Chen, "Response delay optimization in mobile edge computing enabled UAV swarm," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3280–3295, Mar. 2020.
[42] H. Luan *et al.*, "Energy efficient task cooperation for Multi-UAV networks: A coalition formation game approach," *IEEE Access*, vol. 8, pp. 149372–149384, 2020.
[43] P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, "Risk-aware data offloading in multi-server multi-access edge computing environment," *IEEE/ACM Trans. Netw.*, vol. 28, no. 3, pp. 1405–1418, Jun. 2020.
[44] K. Yao, J. Chen, Y. Zhang, L. Cui, Y. Yang, and Y. Xu, "Joint computation offloading and variable-width channel access optimization in UAV swarms," in *Proc. IEEE Global Commun. Conf.*, Taipei, Taiwan, 2020, pp. 1–6.
[45] V. Sdralia, C. Smythe, P. Tzerefos, and S. Cvetkovic, "Performance characterisation of the MCNS DOCSIS 1.0 CATV protocol with prioritised first come first served scheduling," *IEEE Trans. Broadcast.*, vol. 45, no. 2, pp. 196–205, Jun. 1999.
[46] B. Owaidat, H. Nassar, and R. Kassem, "A shortest job first (SJF)-like scheme for efficient call handoff in mobile networks," in *Proc. IEEE 5th Int. Conf. Digit. Inf. Commun. Technol. Appl.*, Beirut, 2015, pp. 216–221.
[47] Y. Zhang *et al.*, "Context awareness group buying in D2D networks: A coalition formation game-theoretic approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12259–12272, Dec. 2018.
[48] Y. Sun, Q. Wu, Y. Xu, Y. Zhang, F. Sun, and J. Wang, "Distributed channel access for device-to-device communications: A hypergraph-based learning solution," *IEEE Commun. Lett.*, vol. 21, no. 1, pp. 180–183, Jan. 2017.
[49] Y. Xu, J. Wang, Q. Wu, A. Anpalagan, and Y. Yao,. "Opportunistic spectrum access in cognitive radio networks: Global optimization using local interaction games," *IEEE J. Sel. Topics Signal Process.*, vol. 6, no. 2, pp. 180–194, Apr. 2012.
[50] Y. Zhang *et al.*, "Context-aware group buying in ultra-dense small cell networks: Unity is strength," *IEEE Wireless Commun.*, vol. 26, no. 6, pp. 118–125, Dec. 2019.
[51] Z. Du, Q. Wu, B. Jiang, Y. Xu, and Z. Qin, "Interference-aware spectrum access self-organization: A weighted graph game perspective," *IEEE Syst. J.*, vol. 12, no. 4, pp. 3250–3259, Dec. 2018.
[52] Y. Xu *et al.*, "Convert harm into benefit: A coordination-learning based dynamic spectrum anti-jamming approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13018–13032, Nov. 2020.

RUNFENG CHEN received the B.E. degree from the College of Communication Engineering, Army Engineering University of PLA, Nanjing, China, in 2018. He is currently working toward the Ph.D. degree with Army Engineering University of PLA. His current research interests include unmanned aerial vehicle communication, mean field game, and collective intelligence.

LI CUI received the B.Eng. and Ph.D. degrees in communication and information engineering from the PLA University of Science and Technology (PLA UST), Nanjing, China, in 2002 and 2009, respectively. Since July 2005, she has been a teacher and researcher with Communication on Engineering Department, PLA UST. Her current research interests include wireless communication and cognitive radio technology.

MENG WANG received the B.S. and M.S. degrees in instructional technology from Nanjing Normal University, Nanjing, China, in 2009. His current research interests include modern educational technology and curriculum and teaching theory.

YULI ZHANG received the B.S. degree from Peking University, Beijing, China, in 2012, the M.S. degree from the PLA University of Science and Technology, Nanjing, China, in 2015, and the Ph.D. degree from the Army Engineering University of PLA, Nanjing, China, in 2018. He is currently with the National Innovation Institute of Defense Technology, Academy of Military Sciences PLA China, Beijing, China. His research interests include spectrum resource allocation, spectrum markets, and game theory.

KAILING YAO received the B.Eng. degree from the School of Communications Engineering, Hohai University, Nanjing, China, in 2014, and the M.E. and Ph.D. degrees from the College of Communications Engineering, Nanjing, China, in 2017 and 2020, respectively. His research interests include game theory and learning algorithms.

YANG YANG received the B.Eng. and Ph.D. degrees in communication and information engineering from the PLA University of Science and Technology (PLA UST), Nanjing, China, in 2005 and 2011, respectively. Since July 2012, she has been a Teacher and Researcher with Communication Engineering Department, PLA UST. Her current research interests include UAV-assisted communication, communication countermeasure, dynamic spectrum sharing, and other wireless communication related topics.

CHANGHUA YAO received the B.S. degree in automation from Zhejiang University, Hangzhou, China, in 2005 and the Ph.D. degree in communications and information systems from the PLA University of Science and Technology, Nanjing, China, in 2016. From 2017 to 2019, she was a Postdoctor with the PLA University of Science and Technology. She is currently a Professor with the School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing, China. Her research interests include unmanned swarm optimization, wireless networks, network security, data analysis, and artificial intelligence.