Research Article

Cooperation-Driven Virtual Terminal Coalition Formation Games for Task Assignment in Mobile Crowdsensing

Haifei Yu, Shiyong Chen, Xiang Liu, and Yucheng Wu

School of Microelectronics and Communication Engineering, Chongqing University, Chongqing, China

Correspondence should be addressed to Yucheng Wu; wuyucheng@cqu.edu.cn

Received 7 June 2021; Revised 6 July 2021; Accepted 16 August 2021; Published 3 September 2021

Abstract

Mobile crowdsensing (MCS) is a popular way of data collection, which forms the large-scale sensing system by smart mobile terminal users and provides multimodal sensor data. In the sensing scenario, there are various sense resource requirements of tasks released by the platform. One of the most urgent issues in MCS is how to choose corresponding users with appropriate sense resources to accomplish assigned tasks. In this article, cooperating among a host of users to perform sense tasks is considered. Firstly, the cooperation among usersto accomplish the sense tasks is described as an overlapping coalition formation game (OCF game). In addition, an initial coalition method of using social networks (SN) is proposed to accelerate the formation of coalition. Finally, the cooperation degree is used to describe the cooperative relationships among users, and virtual terminal coalition formation (VTCF) algorithm is proposed to simplify the process of coalition formation. The simulated results show that the proposed approach effectively improves the system’s utility under the constraints of task cost and sense quality.

1. Introduction

With the development of Internet technology and the popularization of smart devices, diverse high-performance sensors have been embedded into smart terminals, which can monitor and collect surrounding data to promote the development of mobile crowdsensing (MCS). Many new applications have been developed based on smart terminal sensing, such as smart healthcare [1], environment sensing [2], traffic condition detection [3], news [4], and even social networks [5]. Generally, these applications can be divided into three types [6]: personal sense, group sense, and community sense [7]. The mobile crowdsensing, in which a certain number of users are encouraged to participate in smart terminal sensing, is applied to ensure the sense quality of applications.

A typical MCS framework consists of task requesters, sense platforms, and mobile users [8–12]. In MCS, the sense platform assigns tasks required by requesters to the users, and then the matched users complete the tasks through mobile terminals. Finally, users upload the sensed results to the platform. Users upload information collected by sensors to the sense platform via wireless communication technologies such as WiFi and cellular network [13]. With the development of 5G [14], SDN [15], D2D [16], and other communication technologies, the communication in MCS has become faster, and the communication performance has also been greatly improved.

While sensing and uploading data, users need to consume their resources, such as battery energy, computing, and communication. Therefore, unless users are rewarded for compensating for the resources they consume, they may not be interested in sensing tasks. If the platform cannot recruit enough users, then sense quality of the task could not be guaranteed, which will affect the service quality provided by the platform. Hence, some works have considered the incentive mechanism to motivate users participating in the sensing task for the MCS system. The mechanism [17] considers the potential contribution of reverse auction and maintaining existing users when participants recruit new users. The optimization goal is to optimize the composition of users in the system while reducing the system cost. By quantifying the reputation of users [18], an incentive mechanism based on local differential privacy is proposed,
which includes four aspects: incentive, reputation, data disturbance, and data aggregation. The incentive mechanism can select users who can offer more accurate data and compensate users who have spent privacy costs. Simultaneously, the reputation mechanism quantifies the reputation of users to improve the benefits of users.

It is worth pointing out that, during choosing sense tasks, it is assumed that users will make independent decisions and do not know how their strategies affect the sense performance of each task in most literatures. Therefore, most users are likely to participate in a popular task, while other tasks distributed by the platform cannot recruit enough participants, which leads to uneven allocation of users’ resources and a decrease in sensing quality. When the platform releases complex tasks, the user’s limited sense resources may affect the completion of the task.

To prevent the problem of uneven resource allocation mentioned above, cooperating among users is considered. Through cooperation and information exchange, users can understand the specific circumstances of their selected tasks, making more intelligent resource allocation choices. Considering the selfishness and reasonability of users, the coalition game theory is considered as an appropriate method to describe users’ cooperation relationships [19]. However, most of the literatures using coalition formation game for user cooperation assume that one user can only join one coalition [20], which is not suitable for practical scenarios. As each user can participate in multiple sense tasks in the MCS system, the coalitions for different sense tasks may be overlapped.

To tackle the issues discussed above, a novel incentive mechanism that uses social networks to recruit users for the MCS system is described. Users can cooperate through social networks to complete tasks released by the platform. In the process of coalition forming, the concept of cooperation degree inspired by the concept of trust degree [21] is proposed to describe cooperation relationships among users. The main contributions of this paper are summarized as follows:

1. The concept of cooperation degree in the MCS system is introduced to better describe cooperation relationships among users. When the user completes the task, the corresponding cooperation degree will be updated at once, which estimates the positivity of users taking part in the coalition.

2. To accelerate the formation of coalitions, a coalition initialization method based on SN according to the characteristics of users’ social networks is proposed, which makes coalition formation much faster.

3. The task allocation problem in MCS is described as an OCF game. To simplify the process of coalition formation, a virtual terminal coalition formation algorithm (VTCF algorithm) is proposed, which can transform an overlapping coalition problem into a nonoverlapping coalition formation one. Moreover, virtual terminals choose different coalitions to join and allocate sense resources to maximize the system utility. Through simulated experiments, the proposed and reported task allocation algorithms are evaluated and compared. Experimental results have verified the effectiveness of the proposed algorithm.

The rest of this paper is organized as follows. First of all, the related works are reviewed in Section 2; and then system models are described in Section 3. In Section 4, an OCF game is used to describe task assignment of MCS. A coalition initialization method based on SN and the VTCF algorithm are proposed to solve the forming of coalition and the utility of platform is evaluated. In Section 5, the proposed task allocation algorithm is simulated and the effectiveness of the proposed algorithm has been verified. In the end, the related works discussed in this paper are concluded in Section 6.

2. Related Works

At present, some researchers have summarized the research work of task assignment [22–26] and cooperation mechanism [27–30] from different perspectives.

Usually, participants will not voluntarily participate in tasks provided by the sense system. Due to the selfishness of users, the appropriate task allocation and cooperation mechanism are very important to improve the performance of the system. The two complementary modes of opportunistic participation and participatory participation into a two-stage hybrid framework called HyTasker were proposed in [22]. In an offline stage, one group of users (called opportunistic users) is selected, and they complete MCS tasks in daily life. In an online stage, the other groups of users (called participatory users) is assigned; they are required to perform tasks that opportunistic users cannot accomplish. A task management framework called Spatial Recruiter to match staffs to perceptual tasks effectively was proposed in [23]. A greedy heuristic method is designed to select and allocate workers. An optimal incentive mechanism for service providers was studied in [24]. The two-stage Stackelberg game method was used to analyze the participation degree of mobile users and optimal incentive mechanisms of service providers by backward induction. In order to motivate the participants, an incentive mechanism was designed by taking the social network effect, which was from the underlying mobile social field, into consideration. An incentive mechanism algorithm of joint coverage and reputation based on Stackelberg game theory was developed in [25]. In the first place, location information and the historical reputation of mobile users were used to select optimal users, meeting the user’s requirements for information quality. Secondly, the two-stage Stackelberg game was employed to analyze the sense level of mobile users; in this way, an optimal incentive mechanism of the server center was obtained. Last but not least, the expectation maximization (EM) algorithm is used to evaluate the data quality of tasks, and the historical reputation of each user is updated accordingly. An incentive mechanism was proposed in [26], which first selected some users from online social networks as seeds and then social relationships among users
were used to influence selection of potential candidates for task execution. The target was to maximize the number of recruiters and increase coverage based on the initial seed.

A smartphone collaborative sense system to encourage users to participate in sense tasks was considered in [27]. In the case of limited wireless channel resources, the authors therein discussed the incentive mechanism of cooperation and noncooperation and expressed the cooperation of users to complete tasks as overlapping coalitions forming game. Users maximize their own interests by choosing multiple coalitions to join and invest in wireless resources. Compared with noncooperative methods, the system performance will be improved if the cooperative method has been adopted. The cooperation between secondary users (SUs) in cognitive radio to improve sense performance and spectrum efficiency was discussed in [28]. Each SU makes its own cooperative sense decision according to its own business needs. SU's cooperation was described as the formation of overlapping coalitions and proved the stability of the coalition structure. The cooperation strategy between roadside units (RSU) was studied in [29]. Through cooperation, the RSU can coordinate the data types transmitted to the vehicle through the vehicle-to-roadside (V2R) communication link. This solution used the underlying content-sharing V2R communication network increased the diversity of information circulation in the network. Compared with the noncooperative situation, the average rate of return has been improved. A strategy based on a genetic algorithm approach was proposed to achieve optimal collaborative learning groups and the system performance in [30].

3. System Model

In this section, the MCS system model is given, which is composed of a service platform, a wireless base station (BS), and some users carrying smart terminals, as shown in Figure 1. In the system, the sensing platform publicizes \( N \) sense tasks, and the set of sense tasks is denoted by \( N = \{1, 2, \ldots, N\} \). The set of smartphone users is denoted by \( M = \{1, 2, \ldots, M\} \). To simplify the model analysis, it is assumed that a user has only one mobile terminal. The concept of the terminal is equivalent to the user in the subsequent sections. \( M \) mobile terminal users form a social network, represented by graph \( G = (V, E) \); in a \( G \), the vertex \( v \in V \) represents a user on the social network and connects with other users through undirected edges; and the edge \( e \in E \) stands for the communication from its source user to its direct destination user. The weight of the edge indicates the degree of cooperation between the mobile users \( m \) and \( n \) is expressed as \( \cos(m, n) \). If the value of the cooperation degree is larger, it means that the cooperation relationship is more positive.

It is supposed that a mobile terminal user has at most \( r_0 \) types of sense resources and there are \( k \) kinds of sense resources in the MCS system; each user \( i \in M \) is associated with sense resource \( s(i) \), which is a set of sense resources owned by user \( i \), denoted as \( s(i) = \{1, 2, \ldots, r_0\} \); for example, the user has three kinds of sense resources: \( s(i) = \{1, 2, 3\} \). Sense cost of sense resources \( c(i, j) \) represents the cost of user \( i \) providing sense resources \( j \). \( C(i) \) expresses the total cost of user \( i \) providing all sense resources. Neighbor \( n(i) \) is the set of social neighbors of user \( i \), where \( n(i) = \{b| (i, b) \in E\} \), and user \( b \) is the neighbor of user \( i \); for example, the neighbor set of user 1 is \( n(1) = \{2, 3\} \) as shown in Figure 1. Sense quality of sense resources \( q(i, j) \) expresses sense quality of user \( i \) providing sense resource \( j \). The larger value of \( q(i, j) \) means stronger sense ability of user \( i \).

Task \( k \in N \) is denoted by \( \langle R_k, D_k, B_k, U_k, CA_k \rangle \). \( R_k \) expresses a group of sense resources required for task \( k \). \( D_k \) represents the quality requirements of sense resources for task \( k \). \( B_k \) represents the budget of task \( k \). \( U_k \) expresses the revenue of task \( k \). \( CA_k \) expresses the amount of computation required to execute task \( k \). It is supposed that a task always requires multiple users with different sense resources to cooperate with each other. Each user can freely allocate sense resources among these tasks. An \( N \times K \) matrix \( S \) represents resource allocation, in which \( s_{i,j}^k \) denotes that user \( i \) provides the \( j \)-th resource for task \( k \). Resources of the terminal \( i \) are limited as

\[
\sum_{j \in s(i), k \in N} s_{i,j}^k \leq r_0. \tag{1}
\]

The utility function of the certain task, which is a function of terminal sense cost and task budget, is used to evaluate the performance of accomplishing each task. The set of users participating in task \( k \) is denoted as \( M_k \). The utility function of task \( k \) is given by

\[
U_k = \frac{RN(k)}{|R_k|} \times B_k - \sum_{i \in M_k} C(i), \tag{2}
\]

where \( C(i) = \sum_{j \in s(i)} c(i, j) \), \( RN(k) \) is the number of sense resources all users actually provided for task \( k \), and
(RN(k)/|R_k|) is the completion rate of task k. When the user completes the sense task, the cost usually includes consumption of battery energy and computing resources [31]. When the user i performs sense task, the consumption of computing resources and battery energy can be expressed as \( t_i \) and \( e_j \). Let \( f_j > 0 \) denote the computing capability of terminal i in terms of CPU cycles/s. If user i provides sense resource j to execute sense task k, the task completion time is \( t_k = (ca_j/f_j) \), where \( ca_j \) represents the amount of computation required to provide sense resource \( j \) [32], where \( ca_j = (CA_j/|R_k|) \).

To calculate the energy consumption of the terminal performing tasks, the energy consumption model widely used \( e = \kappa f^2 \) for each computing cycle [33], where \( \kappa \) is the energy coefficient depending on the chip structure and \( f \) is the CPU frequency. Therefore, the energy consumption \( e_j \) of terminal i when providing resource \( j \) is calculated as

\[
e_j = \kappa f^2 ca_j.
\]

Therefore, the total cost of user i participating in task \( k \) to provide resource \( j \) is

\[
c(i, j) = \kappa f^2 ca_j + \frac{ca_j}{f_i}.
\]

Considering all tasks of the platform, the utility function of the platform is given by the following formula:

\[
U_{\text{platform}} = \sum_{k \in N} U_k = \sum_{k \in N} \left( \frac{\text{RN}(k)}{|R_k|} \times B_k - \sum_{i \in M_k} \sum_{j \in s(i)} c(i, j) \right).
\]

If users consume their resources without getting any benefits from the sensing platform, they will not be willing to participate in the task. To encourage users to participate in the task and ensure the quality of task completion, an incentive mechanism is designed in which users will be rewarded on the platform according to their contributions to the task. The total revenue of user i can be defined as

\[
P_i = \sum_{k \in N} p_{ik} - C(i),
\]

where \( p_{ik} \) represents the benefit of user i from task k. The total benefit for user i can be described as

\[
\omega \sum_{k \in N} p_{ik} = \omega \sum_{k \in N} U_k,
\]

where \( \omega \) is an incentive factor. Because the benefits of users from different tasks are different due to the resources invested in the task and the utility of the task, the resource allocation among tasks is significant to maximize their benefits and improve the platform’s performance.

### 3.1. Centralized Case

Note that users are not willing to participate in tasks without pay. Therefore, the platform needs to reward the contribution of users, which is often called an incentive mechanism. For an existing MCS system, the platform provides an incentive mechanism, while users try to get maximum profits from the platform. The centralized case represented the upper bound of the system sense performance [27], where users are fully scheduled by the platform. In the centralized case, users are forced to participate in the task without any reward. The platform determines how users participate in the task. This example describes the upper bound of the system’s sense performance, which does not exist in practice because users are unwilling to participate in tasks without any reward.

Given the constraints in (1), the utility of platform in the centralized case is

\[
U_{\text{platform}} = \max_s \sum_{k \in N} U_k,
\]

\[
\text{s.t.} \sum_{j \in s(i)} s_{ij} \leq r_0.
\]

This is a 0-1 integer nonlinear programming (INLP) problem, which can be solved by optimization algorithm [34].

### 3.2. Cooperative Approach

The resource allocation problem of the terminals can be modeled as an OCF game. The terminal has the right to determine the resources allocated to each task. In this scenario, it is supposed that a coalition corresponds to a task and the members of the coalition complete relevant sensing tasks in the OCF game. Moreover, the members of the coalition can allocate sense resources for different tasks. Based on the OCF game model, the overlapping coalition formation algorithm is suitable for multiterminal sense resource allocation. The algorithm can converge to a stable overlapping coalition structure (OCS), improving overall sense performance. The coalition for completing the task can be described as the coalition members and the allocation resources of members.

**Definition 1.** The sense resource allocation problem of terminal is modeled as OCF game, which is expressed as \((M, V)\), where \( M \) represents the set of terminals and \( V \) represents the coalition value; the coalition value of coalition \( C_k \) is defined as

\[
V(C_k) = \frac{\text{RN}(k)}{|R_k|} \times B_k - \sum_{i \in C_k} \sum_{j \in s(i)} c(i, j).
\]

To represent the cooperation behavior among terminals, the cooperation degree between the terminals \( m \) and \( n \) is described as \( \text{co}(m, n) \). The cooperation degree of the coalition is represented by \( \text{Co}(C_k) = \sum_{m, n \in C_k} \text{co}(m, n) \). Assuming that the cooperation degree among all terminals with cooperation relationships is initialized with the same initial value \( \text{co}^{(0)}(m, n) = c > 0 \), among them, the cooperation degree of coalition member \( i \) is expressed as

\[
\text{co}(i) = \frac{\text{Co}(C_k)}{|R_k| \sum_{i \in C_k} p_{ik}} \times g(i) p_{ik}.
\]

where \( \sum_{i \in C_k} p_{ik} \) is the income of all users participating in task \( k \) and \( g(i) \) represents the number of sense resources contributed by terminal \( i \) to the coalition. The update of
cooperation degree among users will be introduced in the third part of Section 4.

In OCF game model, the coalition structure is a set of the total coalitions for accomplishing all tasks and is denoted as \( \prod = \{ C_1, C_2, \ldots, C_N \} \). Each task corresponds to a coalition. Therefore, the size of the coalition structure is fixed, which is equal to \( N \).

When a coalition completes the task, it can be called an effective coalition. The set of effective coalitions is called an effective coalition structure (ECS). In the process of coalition formation, on the one hand, terminal resource allocation needs to maximize the cooperation degree of coalition. On the other hand, it needs to form the coalition with the most significant coalition value. The sense resource allocation problem of all terminals can be formed as the following optimization problem:

\[
U_{\text{platform}} = \max \sum_{i} a \text{Co}(C_i) + \beta V(C_i),
\]

where \( a, \beta \) are scale factors, satisfying \( a + \beta = 1; a, \beta \in [0,1] \) and \( d(k, j) \) represents the sense quality requirement of sense resource \( j \) in coalition \( C_k \). The goal is constrained by three equations: \( R_{C_i} \subseteq \bigcup_{s \in C_i} s(i) \) means that the sense resources provided by the terminal must satisfy sense resources required by sense task \( k \); \( q(i, j) \geq d(k, j) \), \( \forall i \in C_k, j \in s(i) \), means that the quality of the sense resources provided by terminal \( i \) is not lower than the quality of sense resources required by the task, and \( \sum_{i \in C_k} C(i) \leq B_k \) means that cost of coalition \( C_k \) to complete the task does not exceed the budget of the task.

Usually, the coalition pays users a certain reward in some form according to its resource allocation. Hence, a reasonable method of income distribution is very crucial to motivate them to join a coalition. In the OCF game model, the return is a finite vector, expressed by \( P = \{ p_1, p_2, \ldots, p_N \} \). \( P_k \) is the revenue of coalition \( C_k \). In the proposed OCF game, it is assumed that the total profits of the coalition are affected by the participating degree of coalition members. The coalition will offer rewards to its members. For terminal \( i \in C_k \), the revenue of terminal \( i \) from coalition \( C_k \) is defined as follows:

\[
P_{ik} = \omega V(C_k) \frac{\sum_{j \in s(i)} s_{ij}^k}{\sum_{i \in C_k} \sum_{j \in s(i)} s_{ij}^k},
\]

where \( s_{ij}^k \) represents the resources provided by terminal \( i \) for coalition \( C_k \) and \( \sum_{j \in s(i)} s_{ij}^k \) represents total resources provided by terminal \( i \) for coalition \( C_k \). With the revenue that terminal \( i \) obtained from coalition \( C_k \), the terminal \( i \) ‘s total revenue from the platform is

\[
P_i = \sum_{C_k \in \prod} \omega V(C_k) \frac{\sum_{j \in s(i)} s_{ij}^k}{\sum_{i \in C_k} \sum_{j \in s(i)} s_{ij}^k} - C(i).
\]

### 4. Virtual Terminal Coalition Formation (VT CF) Game

Since one coalition corresponds to one task, one terminal can participate in multiple tasks to make full use of the terminal’s resources. Based on distributed social networks, in this section, the coalition forming methods are proposed, where the terminal’s sense cost, sense quality, and the degree of cooperation between terminals are considered to be used to comprehensively express preference relationships in the process of dynamic coalition formation.

#### 4.1. Virtual Terminal

To solve the problem of overlapping coalition formation in an uncertain distributed environment, the concept of the virtual terminal is introduced to reduce the complexity in the overlapping coalition formation process, which is inspired by the concept of virtual auctioneer proposed in [35]. In the actual application of the Internet of things, a terminal usually has multiple sense capabilities. For instance, in the application of air quality monitoring, a terminal can collect CO, NO, and other gas concentrations at the same time. A terminal with multiple sense capabilities is virtualized into multiple virtual terminals. Each virtual terminal has only one sense capability. For example, terminal \( i \) has a sense capability set \( s(i) = \{1, 2, 3\} \), which can be replaced by three virtual terminals \( V(i) = \{v_1^i, v_2^i, v_3^i\} \), and each virtual terminal has only one sense capability. In this case, each virtual terminal uses distributed mobility to form a coalition structure. A terminal can participate in multiple tasks in the coalition structure, while a virtual terminal can only participate in one task. Therefore, overlapping coalition formation is simplified as the problem of nonoverlapping coalition formation, if the concept of the virtual terminal is adopted.

In social networks, each terminal can participate in multiple tasks, which leads to the overlapping coalitions. According to the characteristics of the coalition, the virtual terminal coalition form (VT CF) game is described. The terminal can join multiple coalitions at the same time, while the virtual terminal can only join one coalition. A multisense ability terminal is replaced by multiple virtual terminals with single-sense ability. The virtual terminal participates in the allocation of sense resources. The virtual terminal of terminal \( i \) is represented by \( V(i) = \{v_1^i, v_2^i, \ldots, v_N^i\} \). When \( j \in s(i) \), it indicates that the virtual terminal \( v_j^i \) has sense resources. A distributed coalition formation model for terminals in social networks to ensure convergence is built, which adopts three mobile strategies, exiting, joining, and switching, to form a virtual terminal coalition structure.

#### 4.2. VT CF Algorithm Design

In this section, the resource allocation problem is modeled as the formation of the virtual terminal coalition, in which the virtual terminal participates in the resource allocation process. In the overlapping coalition game, the terminal is willing to cooperate to form a coalition for higher return. The virtual terminal can form a nonoverlapped coalition to complete corresponding tasks. However, each terminal has a variety of sensing capabilities,
causing different tasks. As a result, it is natural to use overlapping coalitions to form a game to simulate resource allocation of terminals with multiple sense capabilities.

In the OCF game, participants form overlapping coalitions to improve the degree of cooperation and coalition value. To improve the cooperation degree of the system, each cooperative terminal can participate in multiple coalitions by using multiple sense capabilities. In this way, terminals form overlapping coalitions to maximize system utility. The introduction of a single-sense virtual terminal instead of a multisense terminal can effectively reduce the complexity of overlapping coalitions to maximize system utility. LZ_hes in-

The variation of cooperation degree between terminals \( m \) and \( n \) can be expressed as

\[
\Delta \text{co}^{(q)}(m, n) = \frac{\text{co}(m)}{\text{degree}(m)} - \frac{\text{co}(n)}{\text{degree}(n)}
\]

(17)

The total cost of all virtual terminals is expressed as \( \sum v'_i \text{sup}(C_k) c(i, j) \). For each coalition structure, the total resource allocation of each terminal \( i \) in each coalition does not exceed its own capacity, as shown in the following equation:

\[
\sum_{C_k \in C} s_{i,j} = s(i), \quad \forall v'_i \in \text{sup}(C_k), j \in s(i).
\]

(14)

4.2.1. The Formation of Initial Coalition Structure. Because the terminal is in social network, the theory of graph search can be used to study the generation of initial coalition structure. This section proposes searching for initial coalition members based on SN, as shown in Figure 2. Task initiator \( a_i \) initiates coalition request and negotiates with its directly connected \( a_j, a_k \) firstly to form a coalition. If sense resources contributed by negotiated coalition still cannot satisfy sense resources required by the task, then negotiate with \( a_j, a_k \)'s acquaintance \( a_l \) to form a coalition. In a large-scale coalition, if the sense cost of existing members is higher than the sense cost of the current member to be negotiated during the negotiation process, the assigned sense resource for the task will be updated, and the process will be repeated until the task requirements are met.

The social network-based coalition initialization method (SN-CI) is described in Algorithm 1. Starting from the task initiator \( i \) to search for the virtual terminal set \( M(C_k) \) and sense resource set \( S(C_k) \) that satisfy sense resource \( R_k \) required by task \( k \) (Steps 6–12), if the sense
resources of task cannot meet the needs of the task, SN is used to find the neighbors of terminal \( i \) to complete the task (Steps 13–31), where \( f s t(Q) \) means that the first terminal in queue \( Q \) is taken out. If sense resource \( j \) contributed by terminal \( i \) can just be satisfied by neighbor \( p \) of terminal \( i \) (Step 17), the cost of neighbor \( p \) providing resource \( j \) is lower than the cost of resource contributed by terminal \( i \). Sense quality requirements are met, and then \( v'_j \) is replaced in the original virtual terminal set \( S(C_k) \) with \( v'_p \) (Step 19). Resource \( j \) in the resource set \( s(p) \) of \( p \) can no longer be used in other tasks, so it is deleted from the queue (Step 20). The resource set \( s(i) \) that previously removed resource \( j \) must readd resource \( j \) (Step 21); otherwise, resource \( q \) is directly added to \( S(C_k) \) (Steps 23–27). \( a d d(Q,n(p)) \) means adding \( p \)'s neighbors to queue \( Q \). The entire social network can be used until all resource requirements of task \( k \) are met (Steps 28–29). Finally, a coalition structure containing all virtual terminal coalition sets is obtained.

Different coalition structures may lead to different system utility. To maximize the utility of the system, it is necessary to find the best coalition structure.

**Algorithm 1: SN-CI algorithm.**

1. **Input:** \( G, M, N \)
2. **Output:** coalition structure \( \Pi \)
3. \( \Pi \leftarrow \emptyset \)
4. for \( k \in N, i \in M \) do
5. \( M(C_k) \leftarrow \emptyset, S(C_k) \leftarrow \emptyset \)
6. for \( j \in s(i) \) do
7. if \( j \in R_k \) and \( q(i,j) \geq d(k,j) \) then
8. \( M(C_k) \leftarrow M(C_k) \cup \{v'_j\} \);
9. \( S(C_k) \leftarrow S(C_k) \cup \{j\} \);
10. \( s(i) \leftarrow s(i) \setminus \{j\} \).
11. end if
12. end for
13. \( Q \leftarrow n(i) \);
14. while \( Q \neq \emptyset \) and \( |S(C_k)| < |R_k| \) do
15. \( p \leftarrow f s t(Q) \);
16. for \( q \in s(p) \) do
17. if \( \exists j \in S(C_k), q \equiv j, q(p,j) \geq d(k,j) \) then
18. if \( c(i,j) > c(p,q) \) then
19. \( M(C_k) \leftarrow M(C_k) \cup \{v'_j\} \);
20. \( S(C_k) \leftarrow S(C_k) \cup \{q\} \);
21. \( s(i) \leftarrow s(i) \cup \{j\} \).
22. end if
23. else if \( q \in R_k \) and \( q(p,q) \geq d(k,q) \) then
24. \( M(C_k) \leftarrow M(C_k) \cup \{v'_p\} \);
25. \( S(C_k) \leftarrow S(C_k) \cup \{q\} \);
26. \( s(i) \leftarrow s(i) \setminus \{q\} \).
27. end if
28. \( Q \leftarrow Q \setminus \{p\} \);
29. \( Q \leftarrow a d d(Q,n(p)) \).
30. end for
31. end while
32. if \( |S(C_k)| = |R_k| \) then
33. \( \Pi \leftarrow \Pi \cup M(C_k) \).
34. end if
35. end for
36. return \( \Pi \).
4.2.2. Virtual Terminal Joining, Exiting, and Switching.

The critical mechanism of coalition formation is to enable virtual terminals to leave or join the coalition according to specified preferences. In particular, each virtual terminal must compare and rank its potential coalitions based on its preferences. Here, the preference order \( \succ_i \) for any virtual terminal \( v_i \) is defined below.

**Definition 2.** (preference order). For any virtual terminal \( v_i \), two coalition structures \( \prod_p, \prod_q \) completing tasks are given. For any virtual terminal \( v_j \), \( \prod_p \succ_i \prod_q \) indicates that \( v_j \) prefers \( \prod_p \) to \( \prod_q \). Since the preference order depends on the system utility, the preference order for the virtual terminal \( v_j \) is defined as follows:

\[
\prod_p \succ_i \prod_q \iff U\left(\prod_p\right) > U\left(\prod_q\right). \tag{19}
\]

According to Definition 2, the utility of the coalition structure increases, and the virtual terminal \( v_j \) will choose this coalition structure. This preference order guarantees a comprehensive increment in the degree of cooperation and platform utility during the formation of the coalition. Using our proposed preference order, the coalition structure can reach stability with a limited number of iterations during its coalition formation.

During the formation of coalition, the virtual terminal decides to join or leave the coalition according to the order of preference. Three-movement rules are defined for possible movement directions of the virtual terminal. Firstly, the virtual terminal exits from the current coalition and enters an idle coalition. Secondly, the virtual terminal leaves an idle coalition and enters any other coalition. Thirdly, the virtual terminal transfers from its existing coalition to a nonidle coalition. Virtual terminals can form different coalition structures with these operations.

**Definition 3.** (exiting rule). It is assumed that the virtual terminals \( v_j \in C_m \) and \( C_m \subseteq \prod_p \). If \( \prod_q \succ \prod_p \), then \( v_j \) leaves coalition \( C_m \) and then coalition structure \( \prod_p \) becomes the coalition structure \( \prod_q \). The new coalition structure is \( \prod_q = \left\{ \prod_p \backslash \{ C_m, C_{\tau + 1} \} \right\} \cup \left\{ C_m \backslash \{ v_j \} \right\} \cup \left\{ C_{\tau + 1} \cup \{ v_j \} \right\} \).

According to Definition 3, if system utility of this newly formed coalition structure is better than that of current coalition structure, the virtual terminal \( v_j \) leaves the current coalition \( C_m \) to join the idle coalition \( C_{\tau + 1} \), and \( \prod_p \) becomes \( \prod_q \). This situation occurs because of virtual terminal \( v_j \), which may have a negative coalition effect. Therefore, \( v_j \) may be more willing to leave coalition \( C_m \) to avoid negative benefits. That is, \( v_j \) is unfriendly, and no other terminal is willing to form a coalition with it.

**Definition 4.** (joining rule). It is assumed that the virtual terminals \( v_j \in C_{\tau + 1} \) and \( C_{\tau + 1} \subseteq \prod_p \). If \( \prod_q \prec \prod_p \) and \( v_j \) join coalition \( C_{\tau + 1} \), the coalition structure \( \prod_p \) becomes \( \prod_q \), and the new coalition structure is \( \prod_q = \left\{ \prod_p \backslash \{ C_{\tau + 1}, v_j \} \right\} \cup \left\{ C_{\tau + 1} \backslash \{ v_j \} \right\} \cup \left\{ C_{\tau + 1} \backslash \{ v_j \} \right\} \).

Based on Definition 4, \( v_j \) does not initially belong to coalition \( C_m \) in the coalition structure \( \prod_p \). It is assumed that the virtual terminal \( v_j \), which does not participate in the sensing task, leaves the current idle coalition \( C_{\tau + 1} \) and joins coalition \( C_m \). The current coalition structure \( \prod_p \) becomes a new coalition structure \( \prod_q \). If the coalition structure \( \prod_q \) is better than \( \prod_p \) for virtual terminal \( v_j \), according to equation (19), \( v_j \) joins the coalition and current coalition structure \( \prod_p \) is replaced by \( \prod_q \). Only when the terminal’s actions improve the effectiveness of the coalition structure will the virtual terminal join a new coalition.

**Definition 5.** (switching rule). It is assumed that the virtual terminal \( v_j \in C_m \) and \( \prod_p \). If \( \prod_p \succ \prod_q \), \( v_j \) is transferred from coalition \( C_m \) to coalition \( C_n \) and coalition structure \( \prod_p \) becomes \( \prod_q \). The new coalition structure is \( \prod_q = \left\{ \prod_p \backslash \{ C_m, C_{\tau} \} \right\} \cup \left\{ C_m \backslash \{ v_j \} \right\} \cup \left\{ C_n \cup \{ v_j \} \right\} \).

According to Definition 5, when a new coalition structure is better than the current coalition structure, \( v_j \) is transferred from the coalition where it was to the new coalition. Switching rules balance scale of different coalitions and improve the utilization of resources. During the formation of the coalition structure, one task can be selected by many virtual terminals, and one virtual terminal can select multiple tasks. When the virtual terminal finds that it can improve its system utility by transferring from coalition \( C_m \) to \( C_n \), it will execute the switch rules. In this way, \( C_m \) can allocate their sense resources to different coalitions autonomously, and the utility of the system will be improved.

**Definition 6.** If, for any \( v_j \in M \), there are \( v_j \in C_m \) and \( v_j \notin C_m \), where \( v_j \) will not leave coalition \( C_m \) and will not join other coalitions, then the coalition structure \( \prod \) of a group of virtual terminals is stable. That is, Nash equilibrium is reached.

According to Definition 6, for a virtual terminal \( v_j \in C_m \) in a stable coalition structure, it will not leave any coalition and will not join any new coalition. Therefore, all virtual terminals will stay in the current coalition, and no changes will be made.

4.2.3. VTCF Algorithm. To achieve a stable coalition structure, the VTCF algorithm is proposed, as shown in Algorithm 2, which is a distributed algorithm and is executed by each \( v_j \in M \). In Algorithm 2, \( v_j \in M \) forms a coalition according to three moving rules. At the beginning of each iteration, \( v_j \in M \) is randomly selected from feasible coalitions. Assuming that \( v_j \) leaves current coalition \( C_m \) and forms a new coalition structure \( \prod_{\text{Quit}} \) (Step 7), judge whether current coalition structure \( \prod_{\text{Quit}} \) is a valid coalition structure (Step 8). If so, using \( v_j \) based on equations (8) and (15), calculate the system utility value of coalition structure \( \prod_{\text{Quit}} \) (Step 9). Similarly, the joining rule and the switching rule can generate a potential new coalition structure. The values of coalition structures \( \prod_{\text{Join}} \) and \( \prod_{\text{Transfer}} \) are calculated by equations (8) and (15) (Steps 15 and 22), respectively. If the coalition structure produced by the roll-out operation is better than the current coalition structure, the coalition structure is updated (Step 11). Similarly, \( v_j \)
considers joining rules (Step 14) as well as switching rules (Step 21) and updates the system utility value of the coalition structure. When a task is completed, the degree of cooperation is updating between terminals (Step 27) through equation (18). \( v_j \) repeats the coalition formation process until all \( v_j \) will not deviate from current coalition or join other new coalitions (Step 29). In other words, the process converges to a stable coalition structure.

**Theorem 1.** After a finite number of iterations, the proposed VTCF algorithm converges to a stable coalition structure.

**Proof.** Assume that the numbers of terminals \( M \) and sense resource types \( r_s \) owned by terminals are limited and there are structural restrictions among terminals. The number of possible coalition structures must be less than \( 2^{Mr_s} \). When the VTCF algorithm is executed, the coalition formation process involves a series of movements of the virtual terminal, which leads to the generation of a series of coalition structures \( \{\prod^{(0)}, \prod^{(1)}, \ldots, \prod^{(r)}\} \), where \( r \) is the total number of movements of virtual terminal. According to equations (10) and (15), a new coalition structure with higher system utility is formed after each virtual terminal moves. In addition, the number of possible coalition structures is limited. Therefore, \( r \) is a finite number. The contradiction proof method is used to prove that if \( \prod^{(r)} \) is final coalition structure after last virtual terminal movement, then it must be stable. Assuming that \( \prod^{(r)} \) is unstable, according to Definition 6, the existence of \( v_j \in M \) causes \( v_j \) to deviate from its current coalition or join a new coalition. Therefore, according to the proposed virtual terminal coalition formation algorithm, \( v_j \) will join, exit, or transfer operations, and coalition structure \( \prod^{(r)} \) will become a new coalition structure. This contradicts the fact that \( \prod^{(r)} \) is the final coalition structure. Therefore, \( \prod^{(r)} \) is a stable coalition structure. Therefore, after a limited number of iterations, the proposed VTCF algorithm enables \( v_j \) to achieve a stable coalition structure.

In the coalition formation process of Algorithm 2, each \( v_j \) seeks to improve value of coalition structure. After each iteration, the movement of \( v_j \) leads to a new coalition structure \( \prod \). Therefore, whenever the coalition structure \( \prod \) changes, more excellent system utility is obtained, and \( v_j \) can adaptively change their cooperation strategy to form a stable coalition structure.

**5. Numerical Simulations**

Task allocation using the VTCF algorithm in a social network environment is simulated. The simulated results of the proposed algorithm are compared with coalition forming based on random initialization, SN-CI, and centralized case [27], respectively. In random allocation, users only randomly
allocate their resources to tasks that they can complete. In the case of SN-CI, the user allocates resources according to the allocation method of Algorithm 1. Finally, the platform utility and the cost of tasks are evaluated in the MCS system. LZ$_{he}$ simulation parameters are shown in Table 1. All curves are generated based on an average of 1000 instances.

In Figure 3, the abscissa is the number of effective iterations, which indicates that a user successfully joins a coalition. It can be seen that the utility of platform becomes stable with the increment of effective iteration numbers based on the VTCF algorithm. Convergence time becomes shorter as the number of users is reduced. The utility value becomes larger when the number of users increases, which indicates that much more users participate in coalition forming. The utility of platform for the VTCF algorithm based on SN and random initialization coalition methods to generate a stable coalition structure are compared in Figure 4. It can be observed that the utility of the platform will increase as the number of iterations increases and tends to be stable. It shows that the initialization coalition method of SN converges faster than random initialization coalition method, and it can accelerate the formation of coalitions.

Figure 5 shows the influence of parameter $\alpha$ on the system utility. The system utility firstly increases with the increment of $\alpha$ and reaches a peak value with $\alpha = 0.4 - 0.6$, which can be regarded as a good combination of cooperation degree and coalition value, and then it decreases as $\alpha$ further increases. Figure 6 shows the effect of the number of tasks on the utility of the platform with $\alpha = 0.5$. By using the VTCF algorithm, the platform utility will increase as the number of tasks increases with $M = 30$. Moreover, the performance of the proposed algorithm is close to that of centralized case and much better than the results of random and SN initialization coalition forming methods.

**Table 1: Simulation parameters.**

| Parameters | Value |
|------------|-------|
| Types of sense resources included in the system: $I$ | 10 |
| Terminal contains the types of sense resources: $R$ | 3 |
| Scale factor: $a, \beta$ | $[0, 1]$ |
| Terminal $t$ provides the quality of $j$ resources: $q(i, j)$ | $[0, 1]$ |
| Budget for task $k$: $B_k$ | $[3, 5]$ |
| Initial cooperation degree: $c$ | 3 |
| Energy coefficient [36]: $x$ | $5 \times 10^{-27}$ |
| CPU capability of each terminal: $f$ | 1–2.8 (GHz) |

Figure 4: The impact of different coalition initialization algorithms on system utility with $M = 30, N = 10$.  

Figure 5: Impact of parameter $\alpha$ on platform utility with $N = 10$.  

Figure 3: The relationship between total number of effective iterations in the VTCF algorithm and system utility with $N = 10$.  

---

10 Mobile Information Systems
The relationship between the total sense cost and the number of tasks is shown in Figure 7. It can be seen that the whole sense cost of platform will increase with the increment of the number of tasks. At the same time, the cost of tasks by using the VTCF algorithm is close to that of centralized case, which indicates that the proposed algorithm consumes less battery energy and computing time. It is very suitable for the IoT applications. In Figure 8, it can be seen that the utility of the platform increases as the number of users becomes larger. This is explained that much more appropriate users join the coalition to improve the utility of the system as the number of users increases. As the budget of the platform is fixed, the coalition forming in the later period will always be concentrated among users with high cooperation and low cost; the system utility will enhance slowly as user numbers furtherly increase.

6. Conclusion

In this article, a cooperation method using overlapping coalition formation game is used to describe the assignment of sense tasks in MCS. The SN-CI based on social networks is proposed to accelerate the formation of coalitions. Moreover, the VTCF algorithm by using the concept of virtual terminals and cooperation degree is proposed to simplify the process of coalition formation. The utility of platform is evaluated by using VTCF, SN-CI, and random, centralized case methods, respectively. The simulated results show that SN-CI method is much quicker than random initialization method in original coalition forming. Furtherly, the utility of platform has been evaluated by the proposed VTCF algorithm and compared with that of using random, centralized case methods. Simulated results have verified the performance of VTCF. The concepts and algorithm proposed in this paper can be regarded as a reference in related study of MCS.

Data Availability

The writing material was derived from different journals as provided in the references. A MATLAB tool has been utilized to simulate the concept.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this study.
Acknowledgments

This research was funded by the National Key Research and Development Program of China under Grant no. 2018YFB2100100.

References

[1] Y. Yang, X. Zheng, W. Guo, X. Liu, and V. Chang, "Privacy-preserving smart IoT-based healthcare big data storage and self-adaptive access control system," *Information Sciences*, vol. 479, pp. 567–592, 2019.

[2] P. Dutta, P. M. Aoki, N. Kumar et al., "Common sense: participatory urban sensing using a network of handheld air quality monitors," in *Proceedings of the 7th International Conference on Embedded Networked Sensor Systems (SenSys 2009)*, pp. 349–350, Berkeley, CA, USA, January 2009.

[3] L. Rui, Y. Zhang, P. Zhang, and X. Qiu, "Location-dependent sensing data collection and processing mechanism in vehicular network," *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 4, 2018.

[4] T. Aitamurto, "The impact of crowdfunding on journalism," *Journalism Practice*, vol. 5, no. 4, pp. 429–445, 2011.

[5] X. Bao and R. Choudhury, "MoVi: mobile phone based video highlights via collaborative sensing," in *Proceedings of the 8th International Conference on Mobile Systems Applications, and Services (MobiSys 2010)*, pp. 357–370, San Francisco, CA, USA, June 2010.

[6] N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. Campbell, "A survey of mobile phone sensing," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 140–150, 2010.

[7] R. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.

[8] Y. Liu, S. Tang, H.-T. Wu, and X. Zhang, "RTPT: a framework for real-time privacy-preserving truth discovery on crowdsensed data streams," *Computer Networks*, vol. 148, no. 15, pp. 349–360, 2019.

[9] B. Zhao, S. Tang, X. Liu, and X. Zhang, "PACE: privacy-preserving and quality-aware incentive mechanism for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 5, pp. 1924–1939, 2021.

[10] B. Zhao, S. Tang, X. Liu, X. Zhang, and W.-N. Chen, "IronM: privacy-preserving reliability estimation of heterogeneous data for mobile crowdsensing," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5159–5170, 2020.

[11] B. Liu, W. Zhong, J. Xie, and L. Kong, "Deep learning for mobile crowdsourcing techniques, methods, and challenges: a survey," *Mobile Information Systems*, vol. 2021, Article ID 5545470, 11 pages, 2021.

[12] Z. Peng, X. Gui, J. An, R. Gui, and Y. Ji, "TDSRC: a task-distributing system of crowdsourcing based on social relation cognition," *Mobile Information Systems*, vol. 2019, Article ID 7413460, 12 pages, 2019.

[13] D. Jiang, W. Wang, L. Shi, and H. Song, "A compressive sensing-based approach to end-to-end network traffic reconstruction," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 507–519, 2020.

[14] S. Yang, X. Qiu, H. Xie, J. Guan, Y. Liu, and C. Xu, "GDSoC: green dynamic self-optimizing content caching in ICN-based 5G network," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 1, p. 3221, 2017.

[15] F. Wang, D. Jiang, H. Wen, and H. Song, "AdaBoost-based security level classification of mobile intelligent terminals," *The Journal of Supercomputing*, vol. 75, no. 11, pp. 7460–7478, 2019.

[16] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1801–1819, 2014.

[17] G. J. Jiang, Z. Yao, B. Zhang, and C. Li, "A reverse auction-based incentive mechanism for mobile crowdsensing," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8238–8248, 2020.

[18] H. Huang, D. Chen, and Y. Li, "ILMDP: incentive mechanism for mobile crowd-sensing based on local differential privacy," *IEEE Communications Letters*, vol. 25, no. 3, pp. 960–964, 2021.

[19] H. Shi, W. Wang, N. Kwok, and S. Chen, "Game theory for wireless sensor networks: a survey," *Sensors*, vol. 12, no. 7, pp. 9055–9097, 2012.

[20] M. W. Baidas and A. B. MacKenzie, "Altruistic coalition formation in cooperative wireless networks," *IEEE Transactions on Communications*, vol. 61, no. 11, pp. 4678–4689, 2013.

[21] L. Militano, A. Orsino, G. Araniti, M. Nitti, L. Atzori, and A. Iera, "Trust-based and social-aware coalition formation game for multihop data uploading in 5G systems," *Computer Networks*, vol. 111, pp. 141–151, 2016.

[22] J. Wang, F. Wang, Y. Wang, L. Wang, Z. Qiu, and D. Zhang, "Hybrid task allocation in mobile crowd sensing," *IEEE Transactions on Mobile Computing*, vol. 19, no. 3, pp. 598–611, 2020.

[23] X. Zhang, Z. Yang, Y.-J. Gong, Y. Liu, and S. Tang, "SpatialRecruiter: maximizing sensing coverage in selecting workers for spatial crowdsourcing," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 6, pp. 5229–5240, 2017.

[24] J. Nie, J. Luo, Z. Xiong, D. Niyato, and P. Wang, "A Stackelberg game approach toward socially-aware incentive mechanisms for mobile crowdsensing," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 724–738, 2018.

[25] J. Zhang, X. Yang, X. Feng, H. Yang, and A. Ren, "A joint constraint incentive mechanism algorithm utilizing coverage and reputation for mobile crowdsensing," *Sensors*, vol. 20, no. 16, pp. 4478, 2020.

[26] J. Wang, F. Wang, Y. Wang, D. Zhang, L. Wang, and Z. Qiu, "Social-network-assisted worker recruitment in mobile crowd sensing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 7, pp. 1661–1673, 2019.

[27] B. Di, T. Wang, L. Song, and Z. Han, "Collaborative smartphone sensing using overlapping coalition formation games," *IEEE Transactions on Mobile Computing*, vol. 16, no. 1, pp. 30–43, 2017.

[28] Z. Dai, Z. Wang, and V. W. S. Wong, "An overlapping coalition game for cooperative spectrum sensing and access in cognitive radio networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 8400–8413, 2016.

[29] W. Saad, Z. Zhu Han, A. Hjorungnes, D. Niyato, and E. Hossain, "Coalition formation games for distributed cooperation among roadside units in vehicular networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 1, pp. 48–60, 2011.

[30] O. Revelo Sánchez, C. A. Collazo, and M. A. Redondo, "A strategy based on genetic algorithms for forming optimal collaborative learning groups: an empirical study," *Electronics*, vol. 10, no. 4, p. 463, 2021.

[31] L. Yang, J. Cao, H. Cheng, and Y. Ji, "Multi-user computation partitioning for latency sensitive mobile cloud applications," *IEEE Transactions on Computers*, vol. 64, no. 8, pp. 2253–2266, 2015.
[32] A. Miettinen and J. Nurminen, “Energy efficiency of mobile clients in cloud computing,” in Proceedings of the 2nd USENIX Workshop on Hot Topics in Cloud Computing, pp. 1–7, Boston, MA, USA, June 2010.

[33] X. Chen, “Decentralized computation offloading game for mobile cloud computing,” IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 4, pp. 974–983, 2015.

[34] B. Di, T. Wang, L. Song, and Z. Han, “Incentive mechanism for collaborative smartphone sensing using overlapping coalition formation games,” in Proceedings of the 2013 IEEE Global Communications Conference (GLOBECOM), pp. 1705–1710, Atlanta, GA, USA, December 2013.

[35] Y. Sun, Q. Wu, J. Wang, Y. Xu, and A. Anpalagan, “VERACITY: overlapping coalition formation-based double auction for heterogeneous demand and spectrum reusability,” IEEE Journal on Selected Areas in Communications, vol. 34, no. 10, pp. 2690–2705, 2016.

[36] T. X. Tran and D. Pompili, “Joint task offloading and resource allocation for multi-server mobile-edge computing networks,” IEEE Transactions on Vehicular Technology, vol. 68, no. 1, pp. 856–868, 2019.