Related Multi-Task Allocation Scheme Based on Greedy Algorithm in Mobile Crowdsensing

Xia Zhuoyue, Raja Kumar Murugesan
School of Computer Science, Taylor's University, Malaysia

***Abstract*—With the popularity of mobile intelligent devices, the mobile crowdsensing (MCS) network based on wireless sensor networks and crowdsourcing technology came into being. There is more and more research on MCS, and it has been applied in many scenarios. Due to the increase in data volume of the MCS platform, the task shows exponential growth. Among them, there will be irreplaceable tasks that belong to the same category, that is, tasks with correlation. If the related tasks can be allocated to the same person for execution, the overhead will be greatly reduced, and the success probability of task allocation will be improved. Firstly, the spatio-temporal distribution of tasks and users is predicted by fuzzy logic to divide spatio-temporal scenarios in this study, and a more suitable multi-task allocation algorithm is selected. Then, when allocating multi-tasks, considering the correlation of tasks, the greedy algorithm is used to allocate multi-tasks according to different scenarios. The experimental results show that compared with the benchmark scheme, the proposed related multi-task allocation scheme based on the greedy algorithm improves the task allocation completion rate by 25.2%, and significantly improves the task allocation success rate in MCS.**

Keywords—Mobile crowdsensing; task allocation; fuzzy logic; greedy algorithm

I. Introduction

With the emergence of intelligent mobile devices and the Internet of Things, mobile crowdsensing (MCS) using wireless sensor networks and crowdsourcing technology has gained a lot of attention. MCS uses mobile devices to collect and process all kinds of sensing data to complete a variety of complex sensing tasks. By using the intelligent mobile terminal as the basic sensing unit, MCS can build a large-scale sensing system that transcends time and space, and complete complicated sensing tasks compared with the traditional sensing technology that relies heavily on professionals and testing tools. As a result, MCS has become a popular new way of sensing. In recent years, MCS has been widely used in environmental monitoring [1],[2], intelligent transportation [3], public safety [4] and health services [5],[6].

The life cycle of MCS can be divided into three parts, namely task allocation, data collection and data aggregation. As the first step in the life cycle of MCS, task allocation is the basis for the successful completion of the whole sensing system. Due to the complexity and diversity of sensing tasks, the completion of a sensing task usually requires the participation of many participants. However, among many participants, each participant has different willingness and proficiency in different sensing tasks, so it is necessary to select the right participant to better complete the task. If the task allocation is unreasonable, it will lead to the poor quality of sensing data or users will refuse to perform sensing tasks, resulting in the wastage of sensing costs and the failure of task sensing. With the in-depth use of the sensing platform, the number of tasks published by the platform has increased rapidly. If the tasks cannot be allocated quickly and reasonably, it will lead to the accumulation of tasks on the platform, low processing efficiency and high task cost, but at the same time, it is difficult for users to obtain matching tasks to execute. Therefore, the quality of task allocation has a significant impact on the quality and cost of sensing task completion.

First, the two cores of task allocation are to consider the order of task allocation and how to choose the executor. The specific implementation of both is related to the temporal and spatial distribution of tasks and related participants. Therefore, the temporal and spatial distribution of tasks and participants is an important factor affecting task allocation. For example, in a time and space, if the distribution of tasks and participants is dense or sparse, it is less difficult to allocate tasks and the completion rate of task allocation is high; in a time and space, if tasks are sparsely distributed and participants are densely distributed, or tasks are densely distributed, and participants are sparsely distributed. In both cases, if the traditional task allocation method is followed, it will lead to unreasonable task allocation and prolong the task allocation period, which will make the allocated tasks unable to be completed within the time constraint and reduce the task completion rate of the whole platform. Therefore, it is very important to determine the temporal and spatial distribution of tasks and participants for the platform to successfully allocate tasks.

Existing research usually uses quantitative statistical methods to count the temporal and spatial distribution of tasks or participants (Citation). Specifically, the spatio-temporal distribution statistics of tasks can be analyzed according to the data collected by the platform most intuitively. However, because the task publishers are free to move, the number of tasks counted by the platform at a certain time point cannot represent the task distribution in a time period. At the same time, in view of the temporal and spatial distribution of participants, the traditional method can realize statistics by asking participants to provide their location information when collecting sensing data. However, this method will not only lead to the leakage of sensing data privacy but also increase the platform overhead. Therefore, how to reasonably determine the temporal and spatial distribution of tasks and participants in MCS is very important. Secondly, the correlation between multiple tasks is also crucial to the success rate of task allocation. We define correlation as that two tasks belong to the same category, but they can't replace each other, so we think they are related. If two tasks belong to the same category or time and space are mutually exclusive, they are considered not relevant. For example, U is a user who cares about his own health. After we allocated task A (collecting exercise data) to user U, task B appeared to collect diet data. Because task B and task A belong to the personal health category, they are related. If task B is also allocated to user U, user U will probably execute task B while executing task A, which will improve the completion rate of task A and task B. Therefore, when allocating multiple tasks, if we can consider the correlation of tasks and allocate related tasks to the same user for execution, it will greatly improve the completion rate of tasks and reduce extra expenses.

At present, with the increase of tasks, multi-task allocation [10]-[16] has become the mainstream research trend of task allocation. However, when allocating multiple tasks, the existing work not only ignores the correlation between the above tasks but also fails to take into account the prediction of the temporal and spatial distribution of users and tasks, which makes it difficult to implement the optimal allocation method in various scenarios and leads to the failure of task allocation.

Aiming at the above problems, this study proposes a related multi-task allocation scheme based on a greedy algorithm. In multi-task allocation, the proposed scheme can determine the temporal and spatial distribution of tasks and users, and based on this, use the correlation between tasks to achieve as many task allocations as possible. The main contributions of this study are summarized as follows:

1) In this study, a task-participant spatio-temporal distribution prediction algorithm based on fuzzy logic is proposed. The uncertain input is processed by fuzzy logic, and the accurate spatio-temporal distribution of tasks and users is obtained, which provides reliable parameter support for subsequent allocation algorithms.

2) Based on the calculated time-space distribution values of tasks and users, combined with the task correlation, we design a related multi-task allocation scheme based on a greedy algorithm, and propose two allocation algorithms for different time-space distributions of tasks and participants in the scheme. Aiming at the situation that the distribution of tasks and participants is balanced or there are enough participants, Algorithm 1 is proposed, which uses correlation allocation to improve the task allocation rate as much as possible; Aiming at the shortage of participants' distribution, Algorithm 2 is proposed. In Algorithm 2, the completion rate of task allocation is improved as much as possible with the help of correlation allocation and the incentive provided by the cost saved by Algorithm 1.

3) Experimental results show that compared with the benchmark scheme, the proposed related multi-task allocation scheme based on a greedy algorithm improves the task allocation completion rate by 25.2%, which significantly improves the task allocation success rate in MCS.

The remainder of this study is organized as follows: Section II introduces the related work of multi-task allocation in MCS. Section III presents the proposed related multi-task allocation scheme based on greedy algorithm. Section IV reports and analyzes the experimental results. Section V concludes the study and discusses directions for future work.

II. Related Work

The core of task allocation is to allocate as many tasks as possible through certain strategies under the condition of satisfying relevant constraints, to obtain sensing data that meets quality requirements.

At present, researchers have put forward a series of task allocation schemes. Among them, the single task allocation is simple, and the key problem at this stage is to find a suitable executor for the task. For example, Zhang et al. [7] propose a blockchain-based hybrid reliable user selection and task allocation scheme for MCS, which achieves decentralized task management through blockchain and smart contracts to enhance system performance. From the user's point of view, An et al. [8] proposed a privacy-preserving scheme for high-quality user recruitment in mobile crowdsensing, which evaluates sensing quality based on data deviation and variance under differential privacy and employs a combinatorial multi-armed bandit to achieve budget-constrained user selection. Wang et al. [9] propose a user recruitment method for sparse mobile crowdsensing that leverages deep nonnegative matrix factorization based on social relationships to identify communities and combines community collaboration with task matching to achieve high-quality sensing data collection and accurate sensing map reconstruction using a small number of users under budget constraints.

With the in-depth study of MCS, it has been applied to more and more fields, which leads to the further expansion of the MCS platform and carries more and more tasks. Therefore, single task allocation can no longer meet the growing sensing needs, and then multi-task allocation appears.

In the task allocation scenario, there are two scenarios: offline task allocation and online task allocation. Different from offline task allocation, online task allocation can't determine the number of participants, so it will affect the distribution quality. Therefore, Yang et al. [10] first predicted the number of participants, and then distributed the platform tasks by using the online task allocation method based on the improved genetic algorithm, to maximize the platform utility and minimize the moving cost of participants. In task allocation, the study of individual task allocation will ignore the characteristics between tasks, such as task distance, task similarity, and task priority level, resulting in poor quality of task allocation. Yin et al. [11] studied a new task allocation and path planning problem in MCS. By considering the routing distance, task similarity, and task priority, a group of task locations is allocated to suitable workers, and the location access order of individual workers is determined to realize the overall rationality of the maximum location sequence. Previous studies paid more attention to the completion rate of task allocation or the minimum cost of task allocation, while ignoring the individual situation of platform staff. Because each participant's sensing willingness, knowledge level, and credibility are different, Rahman et al. [12] put forward a new quality-aware personalized task matching framework, which can better match tasks with participants.

For multi-task allocation, the sensing will of the participant is also very important. Because of the difference in users' sensing ability, the sensing result will be inaccurate. To overcome these problems, Ji et al. [13] proposed a comprehensive multi-task allocation model. Based on the three constraints of the total task budget, perceiving the quality and workload of a single task, the proposed large-scale evolutionary algorithm, a specific problem-solving strategy, and a new genetic operator, multi-task allocation can get as many feasible tasks as possible. In order to recruit suitable users for heterogeneous tasks, Ma et al. [14] proposed three user recruitment algorithms based on greed to solve the problem of heterogeneous user recruitment for multi-heterogeneous tasks, thus minimizing the total platform payment and maximizing the task coverage. In task allocation, users will also be unwilling, or objective conditions will not allow them to move to the task area. Gao et al. [15] proposed a UAV-assisted multi-task allocation method. This method uses drones to verify the data collected by users and uses drones to collect data in data areas not covered by human participants to optimize the sensing coverage and data quality of multi-task allocation. When MCS is applied in an energy transportation system, there are some challenges, such as a single point of failure, low efficiency of independent task allocation, and inability to deal with safety emergency tasks in time. In order to solve these problems, Li et al. [16] defined a decentralized ITS architecture based on blockchain and proposed a concurrent task and safety emergency task allocation method based on reinforcement learning, which can maximize the utility of concurrent tasks on the basis of meeting the requirements of safety emergency tasks.

At present, the focus of multi-task allocation is on multi-task allocation under time constraints [17]-[21]. Li & Zhang [17] studied the multi-task allocation under time constraints, mainly considering the leisure time owned by users and the due time of each task. Huang et al. [18] focus on the execution time of tasks, which solves the problem that mobile users with a limited time budget undertake multiple sensing tasks. Under the limited time limit, the optimal solution can be obtained through the multi-task framework by selecting the participants who have completed the tasks the most times and the shortest total travel distance. In time-limited multi-task allocation, existing research tends to assign most tasks to users with high reputation, which is unfair to new users and leads to low efficiency of task execution. To overcome these problems, Shen et al. [19] proposed a constrained multi-objective optimization model for variable-speed multi-task allocation. Through the three-stage multi-objective mixed shuffling frog leading algorithm, the user returns to the maximum extent and the task completion time is minimized.

In addition, with the complexity of sensing scenes and the diversification of time constraints, taking time constraints as a single auxiliary constraint can no longer achieve a good task allocation scheme. Wei et al. [20] proposed a semi-conditional moving sensation (SO-MTTA) scheme to solve the task allocation problem with multiple time constraints, to maximize the sensing value obtained by the platform. Meitei & Marchang [21] proposed a greedy task allocation method based on interval partition to maximize the profit of the platform in a time-dependent sensing environment.

With the deepening of research, spatial constraints have also become an important research point. Most scholars consider the overall distribution effect when considering the meaning of multitasking. Although the constraints of overall execution time or overall sharing are minimized or met, respectively, according to the distribution results, it is possible that the tasks published are not effectively executed for each task publisher. Wang et al. [22] conducted extensive research on this issue and suggested using the task framework to define the threshold for measuring effective task execution. Only tasks that exceed the threshold are considered as valid tasks. Specifically, the system uses space-time coverage to measure tasks. The task is regarded as a low-quality execution, and if the coverage of time and space is below the threshold, it will not improve the overall task execution quality. The framework can effectively identify tasks suitable for workers and ensure the best performance of each task through the above methods. Liu et al. [23] proposed a privacy-preserving task allocation method for mobile crowdsensing that protects users' temporal and spatial information while maximizing worker income. Ye et al. [24] studied a new task allocation problem in a multi-center supply chain environment with multiple distribution centers. In order to solve this problem, a task allocation framework based on geographical division is proposed. The first stage is the geographical division stage, and the second stage is the task allocation stage. This method can effectively maximize the total number of allocated tasks and minimize the difference in the average number of allocated tasks. To solve the problem of interruption in the task allocation process, wang et al. [25] proposed a method based on the allocation graph to solve the problem of RoBust task allocation (RBTA), which can reduce the cost of workers' detour and minimize the robustness of the allocation scheme.

Fuzzy logic has been applied to the field of mathematics in theory. With the development of computer science and artificial intelligence, it needs to be applied to industrial control systems. Such progress requires computers to perform based on human cognition. Fuzzy logic has been widely incorporated into the computer field.

For example, by extending the standard Mamdani fuzzy logic controller, an expert system based on fuzzy logic to diagnose patients with possible heart disease was established [26]. With regard to bringing fuzzy logic into the proofreading system, some dough was made under the supervision of a fuzzy control system. According to different initial temperatures, different amounts of yeast were added to the dough. The controller will make judgments and adjustments in the whole process to control the volume of dough. In this case, the fuzzy logic controller provides optimal control and improved interference suppression characteristics without a mathematical model [27]. Wang et al. [28] put forward a spatio-temporal model based on a transformer, which uses spatio-temporal relationships to infer and predict sparse sensing data. By using short-term perceived data to predict long-term data, not only is the data cost reduced, but also the data trend can be identified, which effectively improves the working effect of the model.

Yang et al. [29] used fuzzy logic to predict the distribution of users and solved the multi-task allocation problem under time and space constraints through a hybrid greedy algorithm. Zhang et al. [30] put forward a multi-task allocation method based on liquidity prediction, which uses fuzzy logic to analyze historical data to predict the movement of workers, so that workers can better match tasks and complete tasks as much as possible.

Although the above technologies show superior performance in some aspects, they only consider some temporal or spatial characteristics of tasks and participants, and also lack comprehensive analysis and modeling of participant density in time and space. Moreover, the above research did not consider the correlation between multiple tasks. If two related tasks are allocated to the same participant, it will greatly save time of task execution. On the contrary, if the related tasks are allocated to different executors, it will lead to a waste of time and resources. The existing research on multi-task allocation does not consider the correlation between tasks, which may lead to the failure of task allocation or the difficulty of matching tasks with the most suitable participants, resulting in waste of resources and low task completion rate.

Therefore, this study fully considers and analyzes the influence of task distribution and participant distribution on task allocation through the fuzzy logic control method and studies the task correlation in multi-task allocation under time and space constraints in order to achieve the maximum task completion rate.

III. Methods

A. System Model

The realization of MCS is based on the sensing platform, on which task publishers and task participants complete the corresponding cooperation according to their own wishes and the task allocation strategy of the platform, as shown in Fig. 1. Specifically, there are the following steps:

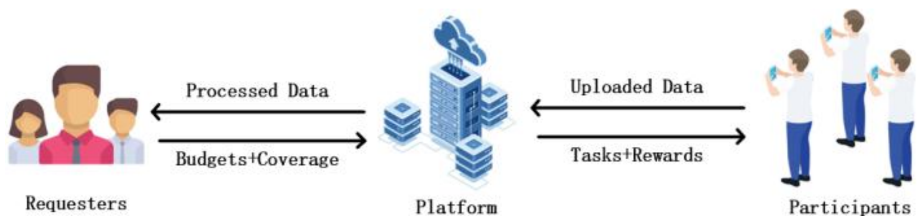


Fig. 1. The illustration of task allocation.

First, the task publisher will publish the task according to its own needs and hand it over to the platform. The submitted data contains the specific information on the task and the rewards that the participants are willing to pay after completing it. Secondly, after receiving the task information, the platform will sort it out and publish it for the participants to check; subsequently, the participants will release their willingness to execute to the platform. The platform will select the appropriate participants to perform the task according to the execution time, implementation place and user's willingness to perform the task. The selected participants will choose whether to perform the task according to their own situation and the task reward. Finally, participants who perform tasks are selected to collect sensing data and upload them to the sensing platform. Thus, a complete sensing task is completed.

The scheme of this study mainly includes two parts. The first part is to determine the temporal and spatial distribution of tasks and participants. The second part is to allocate tasks according to the temporal and spatial distribution of tasks and participants, combined with the correlation between multiple tasks.

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or headings unless they are unavoidable.

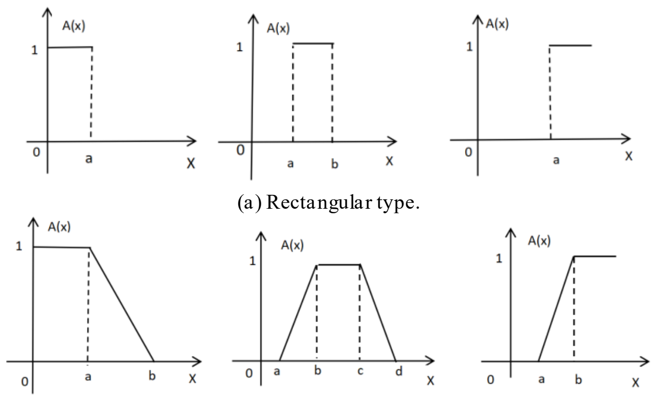
B. Fuzzy Logic

In the traditional description of temporal and spatial distribution, some non-quantitative words are usually used to describe the distribution of tasks or participants, such as "more", "a little less" and "sparse". In daily communication, because it is only necessary to express the meaning correctly, such a description can make the two sides understand each other's meaning. But in the computer world, machines can't understand these non-quantitative words, and what computers can understand must be a specific and accurate numerical value. Therefore, we need to introduce fuzzy logic to solve the problem of semantic uncertainty.

Fuzzy logic mainly includes the following three steps, namely fuzzification, fuzzy reasoning and defuzzification. The so-called fuzzification is to process our specific inputs and get a fuzzy set of inputs. Secondly, according to the fuzzy set output in the first step, combined with fuzzy rules, an output fuzzy set can be obtained, which is fuzzy reasoning. The third step is defuzzification, and the fuzzy conclusions are transformed into concrete and clear output values.

1) Fuzzy set: The definition of a fuzzy set can decompose an input into the membership degree of each part in the set. For example, we define the fuzzy set of rain sizes as "heavy rain, moderate rain and light rain". Then, for 20 ml of rainfall, the membership degree of light rain is 0.3, the membership degree of moderate rain is 0.5, and the membership degree of heavy rain is 0.2.

2) Membership function: The most used membership functions are rectangular and (semi) trapezoidal functions, which are divided into small, large and intermediate types. The specific membership function expressions are shown in Fig. 2.



(a) Rectangular type.

(b) Trapezoidal type.

Fig. 2. Common membership function diagram.

3) Fuzzy rules: After fuzzy input, we need to construct rules and use the operation of fuzzy logic to combine the obtained membership degrees as the basis for decision-making. The decision rules of fuzzy logic are sets containing a series of logical statements, which have the following structure:

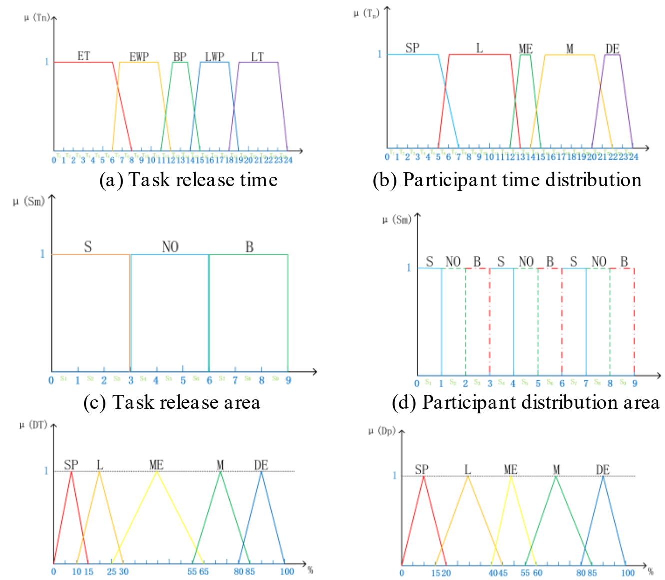
IF <antecedents> THEN <consequent>

4) Defuzzification: In fuzzy logic, we convert input values into membership degrees of each set through fuzzification and then get some effective output values through fuzzy rules. However, having only a few output values is not helpful for problem analysis, so we need to transform some fuzzy output values. Therefore, the process of converting some fuzzy output values into accurate values is called defuzzification.

C. Temporal and Spatial Distribution of Tasks and Participants

For the definition of the temporal and spatial distribution of tasks and participants, the two core influencing factors are time and space. For example, during working hours and in densely populated areas, the amount of task release is relatively large. However, in the rest time and sparsely populated areas, the amount of task publishing is relatively rare. Therefore, with the help of fuzzy logic, we define the input as time and space. In view of time, this scheme divides 24 hours a day into l parts, which are recorded as $T_L = \{T_1, \ldots, T_n, \ldots, T_l\}$. For space, this scheme divides a region into p sub-regions, which are denoted as $S_P = \{S_1, \ldots, S_m, \ldots, S_p\}$. Therefore, we can use time and space as the input of fuzzy logic to calculate the temporal and spatial distribution of tasks and participants. As the output of fuzzy logic, we define the spatio-temporal distribution parameters of tasks as D_T and the spatio-temporal distribution parameters of participants as D_P.

First, we need to determine the fuzzy set of the spatio-temporal distribution of tasks and the spatio-temporal distribution of participants. We define the fuzzy set of task release and participants' participation time periods as early trough (ET), early wave peak (EWP), balance period (BP), late wave peak (LWP) and late trough (LT). Then, the fuzzy sets of the sub-regions where the tasks and participants are located are defined as small (S), normal (NO) and big (B). As for the output variables, the fuzzy set of the spatio-temporal distribution of tasks and participants is defined as sparse (SP), less (L), medium (ME), more (M) and dense (DE).



(a) Task release time (b) Participant time distribution

(c) Task release area (d) Participant distribution area

(e) Distribution of tasks (f) Distribution of participants

Fig. 3. Membership function diagram.

Referring to common models, we divide a day into 24 equal parts, so l=24, $T_L = \{T_1, \ldots, T_n, \ldots, T_{24}\}$. Divide an area into 12 sub-areas, p=12, $S_P = \{S_1, \ldots, S_m, \ldots, S_{12}\}$. According to experience, we can get the membership function curve in Fig. 3.

Because the input conditions of this scheme are time and space, and the output is a time-space distribution, the rules can be defined by fuzzy reasoning. Specifically, the rules defined in this scheme are shown in the following table:

TABLE I. FUZZY LOGIC RULE TABLE

S_m \ D_t/D_p \ T_n	ET	EWP	BP	LWP	LT
S	SP	ME	L	ME	SP
NO	L	M	ME	M	L
B	ME	DE	M	DE	ME

According to the rules in Table I, if the task release period is in the early trough and its area is a partition with few tasks, it can be obtained that the temporal and spatial distribution of tasks is sparse. Through the IF-THEN rule, we can express this rule as: IF (T_n = ET) and (S_m = S), THEN (D_T = SP). Similarly, if the movement period of participants is at the late wave peak, and the area where they are located is a partition with many tasks, it can be obtained that the spatial and temporal distribution of participants is dense. The above rules can be expressed as: IF (T_n=LWP) and (S_m= B), THEN (D_P = DE). For the calculation rule "and" in the above fuzzy logic, this study adopts the principle of minimum membership.

Based on the above work, this study uses the weighted average method to achieve defuzzification, and the specific formula is as follows:

$$\mathrm{OT}=\frac{\sum_{1}^{i}\mathrm{u}(\mathrm{D_i})*\mathrm{OW_i}}{\sum_{1}^{i}\mathrm{u}(\mathrm{D_i})} \quad (1)$$

Among them, $u(D_i)$ represents the value of the membership function D and the weight OW_i, usually taking the middle value of each set. Through Formula (1), we can convert some fuzzy outputs into a usable, accurate output value.

We will illustrate it with examples. Assume that the two input values of the task fuzzy logic method are time point 7.5 and sub-area 4. Firstly, we can get IF (T_n= ET) and (S_m= S), THEN (D_T= SP), IF (T_n= EWP) and (S_m= S), THEN (D_T = ME) by consulting the rule diagram. Then the fuzzy set is calculated by using the membership function of these two inputs. It can be obtained that the value of the early trough T_n(ET [7.5])= 0.25, the value of the early wave peak T_n(EWP [7.5])=0.75, and the value of the member function of the small amount S_m(S [4])= 1. Because the conditional parts of two fuzzy logic rules are connected by the AND method, the value of the membership function will be calculated by the minimum value of the corresponding membership functions. For the rules IF(T_n= ET)and(S_m= S), THEN(D_T= SP), the value ofD_T= SP can be calculated as min(0.25, 1)= 0.25 by the minimum membership rule, and for the same rules IF(T_n= EWP)and(S_m= S), THEN(D_T= ME), the value of D_T= ME can be calculated as min(0.75, 1)=0.75 by the minimum membership rule. Then the clear value is 35.625 by defuzzification with the weighted average method.

Similarly, we assume that the two input values of the participant's fuzzy logic method are time point 5.5 and sub-region 2. Firstly, we can get IF (T_n= ET) and (S_m= NO), THEN (D_P= L), IF (T_n= EWP) and (S_m= NO), THEN (D_P = M) by consulting the rule diagram. Then the fuzzy set is calculated by using the membership function of these two inputs. It can be obtained that the value of early trough T_n(ET [5.5]) = 0.75, the value of early wave peak T_n(EWP [5.5]) = 0.25, and the value of normal (NO) member function S_m(NO [2]) =1. Because the conditional parts of two fuzzy logic rules are connected by the AND method, the value of the membership function will be calculated by the minimum value of the corresponding membership functions. For the rules IF (T_n= ET) and (S_m= NO), THEN (D_P= L), the value of the membership function D_P= L can be calculated as min (0.75, 1) = 0.75 through the minimum membership rule. Similarly, for IF (T_n= EWP) and (S_m=NO), THEN (D_P= M), the value of membership function D_P=M can be calculated as min (0.25, 1) = 0.25 through the minimum membership degree rule. Then the clear value is 40 by defuzzification with the weighted average method.

D. Multi-Task Allocation Scheme

To facilitate the following description, we first explain the parameters used:

In time T_n and space S_m, this scheme defines task sets as Task={t_1, t_2, ..., t_i}, participant sets as Part ={p_1, p_2, ..., p_j}, and the compensation that the task publisher is willing to pay for task sets is B_i={b_1, b_2, ..., b_i} respectively, and the cost of participant for each task is $c_{i,j}$ [see Formula (2)]:

$$C_{i,j}=\begin{bmatrix} c_{1,1} & \cdots & c_{1,j} \\ \cdots & & \cdots \\ c_{i,1} & \cdots & c_{i,j} \end{bmatrix} \quad (2)$$

To describe the participants' willingness to perform tasks, this scheme defines $PW_{i,j}$ as the sensing willingness of participant p_j to perform task t_i, and the parameter $PW_{i,j}$ is binary. When $PW_{i,j}$=1, the participant is willing to perform the sensing task; otherwise, $PW_{i,j}$=0, the participant is unwilling to perform the task.

In addition, we define the parameter TC(t_i,t_j) to describe the correlation between tasks. In detail, if task $t_i$and task t_j belong to the same category and time-space, and the participants can perform task t_j while performing task t_i, it is considered that they are related, and TC (t_i, t_j) = 1; on the other hand, it is considered that tasks t_i and t_j are irrelevant, and the correlation coefficient TC (t_i, t_j) = 0. For example, task 1 is to collect automobile acceleration data, and task 2 is to collect automobile driving recorder data. Task 1 and task 2 belong to automobile data and can be executed in the same time and space, but they are irreplaceable, so task 1 and task 2 are related.

When performing sensing tasks, because the participants have moved to a certain time-space area when performing task t_i, if task t_j is related to task t_i, at this time, for users, performing task t_j does not need additional space movement, so compared with simply performing task t_j, it can save a certain time-space cost. At the same time, because task t_j and task t_i belong to the same kind of tasks, participants can perform them at the same time without secondary learning, so the execution cost of participants will be further reduced and the willingness to perform will be enhanced. We define the coefficient as a way to calculate the cost saved by users in performing related tasks. If a user has already allocated a task t_i and then allocated a related task t_j, the user's execution cost will be reduced from c_j to $a*c_j$($0<a<1$). At this time, because of the correlation allocation, we define the funds saved by the correlation allocation as the balance expenditure, which is expressed by R.

In addition, in the process of allocating tasks, some tasks are urgent, and some tasks are not so urgent. To distinguish the urgency of the task, we define the parameter ε to describe the urgency of the task. The greater the parameter ε, the more urgent the task is, and it needs to be allocated to the participants as soon as possible. On the contrary, it means that the urgency of the task is average and can be allocated later.

1) Multi-task allocation algorithm: Based on the fuzzy logic calculation before, we can get the time and space distribution of tasks and participants in a certain time and space interval. In a time T_n and space S_m, if the spatio-temporal distribution $D_T$of tasks is greater than or equal to the threshold β_P, and the spatio-temporal distribution D_P of participants is greater than or equal to the threshold β_T, or the spatio-temporal distribution D_T of tasks is less than the threshold β_T, then we think that the resources of tasks and participants can match in this area, which is defined as case 1. If the spatio-temporal distribution D_T of tasks is greater than or equal to the threshold β_T, and the spatio-temporal distribution D_P of participants is less than the threshold β_T, it means that there are more tasks and fewer participants in this spatio-temporal area, and it is necessary to provide some incentives for participants to encourage them to complete tasks as much as possible, which is defined as case 2. However, if the temporal and spatial distribution D_T of tasks is less than the threshold β_T, and the temporal and spatial distribution D_P of participants is greater than or equal to the threshold β_P, it means that there are few tasks and many participants in this area, and there is no need to provide additional incentives for participants, which is defined as case 3.

It can be observed that for the above cases 1 and 3, either the number of participants and the number of tasks are balanced, or the number of tasks is large and the number of participants is small. In both cases, we can adopt a general distribution strategy and do not need additional incentives to motivate users to perform tasks. Therefore, we classify cases 1 and 3 into the first situation. In case 2, there are more tasks and fewer participants. To enable users to complete tasks as much as possible, extra incentives should be given to users. Therefore, we regard case 2 as the second kind of situation. The task allocation strategy in each case is as follows:

We divide the Task set into the first kind of case, task set $Task_1$ and the second kind of situation, task set $Task_2$. The allocated task set is defined as TA, and the initial value is ta = $\emptyset$.

For the first kind of situation: firstly, we sort the tasks $Task_1$ according to the urgency from high to low and find out the task t_{i1} with the highest urgency. Among all users, we screen out the users who are willing to sense the task t_{i1}, and among these willing users, we find out the user p_{j1} who has the lowest sensing cost of t_{i1}. If the cost of executing task t_{i1} by user p_{j1} is less than or equal to the offer that task t_{i1} is willing to pay, task t_{i1} is allocated to user p_{j1} for execution. Then, firstly, the tasks that user $p_{j1}$is willing to perform are screened out from the task list, and among these tasks, the related task t_{i2} of task t_{i1} is found according to the urgency from high to low. At this time, if the cost a*$c_{i2,j1}$ of allocating the task t_{i2} to the user p_{j1} is less than or equal to the minimum cost $Minc_{i2}$=min($c_{i2,1}$,...,$c_{i2,j}$) of the task t_{i2} being executed by all users, and a*$c_{i2,j1}$ is less than or equal to the cost that the corresponding publisher is willing to pay for the task t_{i2}, then the related task is allocated and the task t_{i2} is allocated to the user p_{j1} to be executed together.

In the process of allocation, we define R as the saving cost of task allocation in the first case, then $R=\sum_1^n(1-a)*c_{i,j}$, where n is the number of times to complete the related task allocation.

Algorithm 1 gives the task allocation process in the first situation.

Algorithm 1: Related multi-task allocation algorithm in the first situation based on the greedy algorithm.

Input: Task, Part, B_i, $C_{i,j}$, $PW_{i,j}$, TC（t_i，t_j）,ε,a
Output: TA, R

//In the first case, assigning $Task_1$ to participants.
R=0
Rank tasks in $Task_1$ by ε
Repeat
for $t_i \in Task_1$, i from 1 to i , do
  the max(ε) task t_i from $Task_1$ call t_{i1}
  select all p_j when $PW_{i1,j}$=1 call P
  find min($C_{i1,j}$) p_j from P call p_{j1}
  if min($C_{i1,j1}$) $\le B_{i1}$ then
    task t_{i1} is allocated to participant p_{j1}
     $Task_1$=$Task_1$-t_{i1}, TA=TA+t_{i1}
     select all task t_i from $Task_1$ when $PW_{i,j1}$=1 call T
     find the max(ε)task t_i from T when TC（t_{i1}，t_i）=1 call t_{i2}
     If a*$C_{i2,j1}$ $\le$ min($C_{i2,j}$) & a*$C_{i2,j1}$ $\le B_{i2}$ then
     task t_{i2} is allocated to participant p_{j1}
     $Task_1$=$Task_1$-t_{i2}, TA=TA+t_{i2}
     R=R+(1-a)*$C_{i2,j1}$
     else
       continue
  else
     Continue
return TA, R

For the second kind of situation: firstly, find out the task t_{i3} with the highest urgency in the task set $Task_2$, screen out the users who are willing to sense the task t_{i3}, and find out the user p_{j2} who has the lowest sensing cost for the task t_{i3} among these willing sensing users, and if the cost for the user p_{j2} to execute the task t_{i3} is less than or equal to the reward paid for the task t_{i3}, allocate the task t_{i3} to the user p_{j2} for execution; If the cost $C_{i3,j2}$ of user p_{j2} executing task t_{i3} is greater than the offer b_{i3} that task t_{i3} is willing to pay, and it still needs R_{i3}=$C_{i3,j2}$-b_{i3} funds to distribute the task, then the remaining balance cost R'=R-R_{i3} at this time, if $R' \ge 0$, incentive R_{i3} is carried out to distribute task t_{i3} to user p_{j2} for execution.

When the task t_{i3} is allocated to the user p_{j2} for execution, the tasks that the user p_{j2} is willing to execute are first screened out in the task list, and then the related task t_{i4} of the task t_{i3} is found in the order of urgency. At this time, if the cost a*$c_{i4,j2}$ of allocating task t_{i4} to user p_{j2} is less than or equal to the minimum cost of task t_{j4} being executed by all users $Minc_{i4}$=min($c_{i4,1}$,...,$c_{i4,j}$), and a*$c_{i4,j2}$ is less than or equal to the cost paid by the task publisher to task t_{i4}, relevant tasks are allocated, and task t_{i4} is allocated to user p_{j2} to be executed together. At this time, the remaining cost is $R'=R'+(1-a)*c_{i4,j2}$; if the cost $a*c_{i4,j2}$ of allocating task t_{i4} to user p_{j2} at this time is less than the minimum cost of task t_{i4} being executed by all users $Minc_{i4}=\min(c_{i4,1},...,c_{i4,j})$, and $a*c_{i4,j2}$ is greater than the cost b_{i4} that task publisher is willing to pay for executing task t_{i4}, the extra incentive required at this time is $R_{i4}=a*c_{i4,j2}-b_{i4}$, and the saving cost is $R''=R'-R_{i4}$. If $R''\geq 0$, the incentive is carried out and task t_{i4} is allocated to user p_{j2} for execution.

Repeat the above process until all tasks are allocated or the number of users is exhausted.

Algorithm 2 gives the task allocation process in the second situation.

Algorithm2: Related multi-task allocation algorithm in the second situation based on the greedy algorithm.

Input: Task, Part, B_i, $C_{i,j}$, $PW_{i,j}$, TC（t_i，t_j）,ε,a,R

Output: TA

//In the second case, assigning $Task_2$ to participants.
Rank tasks in $Task_2$ by ε
Repeat
for $t_i \in Task_2$, i from 1 to i , do
the max(ε) task t_i from $Task_2$ call t_{i3}
select all p_j when $PW_{i3,j}$=1 call P'
find $\min(C_{i3,j})$ p_j from P' call p_{j2}
if $\min(C_{i3,j2}) \leq B_{i3}$ then
task t_{i3} is allocated to participant p_{j2}
$Task_2=Task_2-t_{i3}$, $TA=TA+t_{i3}$
else
$R_{i3}=C_{i3,j2}-b_{i3}$
$R'=R-R_{i3}$
if $R'\geq 0$ then
provide incentive R_{i3} for task t_{i3}
task t_{i3} is allocated to participant p_{j2}
$Task_2=Task_2-t_{i3}$, $TA=TA+t_{i3}$
$R=R-R_{i3}$
else
continue
if task t_{i3} is allocated to participant p_{j2} then
select all task t_i from $Task_2$ when $PW_{i,j2}$=1 call T'
find the max(ε)task t_i from T' when TC（t_{i3}，t_i）=1 call t_{i4}
If $a*C_{i4,j2} \leq \min(C_{i4,j})$ & $a*C_{i4,j2} \leq B_{i4}$ then
task t_{i4} is allocated to participant p_{j2}
$Task_2=Task_2-t_{i4}$, $TA=TA+t_{i4}$
$R'=R'+(1-a)*c_{i4,j2}$
else
$R_{i4}=a*c_{i4,j2}-b_{i4}$
$R''=R'-R_{i4}$
if $R''\geq 0$ then
provide incentive R_{i4} for task t_{i4}
task t_{i4} is allocated to participant p_{j2}
$Task_2=Task_2-t_{i4}$, $TA=TA+t_{i4}$
$R'=R'-R_{i4}$
else
continue
else
Continue
return TA

IV. Results

To illustrate the performance of the proposal, this section compares the proposed algorithms with the baseline algorithm derived from the existing work [29]. Note that the core idea of [29] is allocating sensing tasks by means of a greedy algorithm. Therefore, the classical greedy algorithm is adopted as the baseline for comparison.

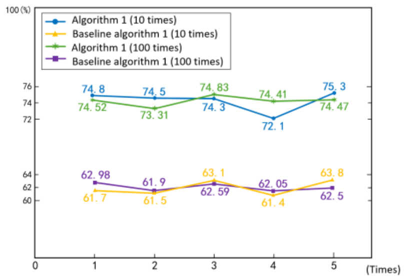
A. Task Allocation Rate

We set the discount coefficient a=0.5, the urgency ε is randomly generated between 1 and 10, the participants' wishes $PW_{i,j}$ and the correlation coefficient TC（t_i，t_j） are randomly assigned between 0 and 1, the compensation B_i that the task publisher is willing to pay for the task is randomly generated between 1 and 15, and the cost $C_{i,j}$ that the user i needs to perform the task j is randomly generated between 5 and 20.

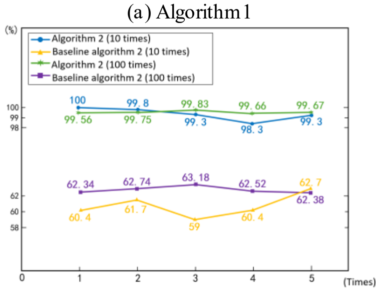
Task set of algorithm 1 contains 100 randomly generated Tasks, set Part contains 50 randomly generated participants, task set of algorithm 2 contains 100 randomly generated tasks, and set Part contains 20 randomly generated participants. In the process of testing, we tested two groups of data respectively. The first group of data was tested 10 times, each time, with five times. The second set of data is tested 100 times at a time, with five times.

As can be seen from Fig. 4, in the first set of data, the average task allocation rate of Algorithm 1 is 74.24%, that of Algorithm 2 is 99.34%, and that of Algorithm 1 and 2 is 86.77%. At the same time, the average algorithm allocation rates of baseline Algorithm 1 and baseline Algorithm 2 are 62.3% and 60.84%, respectively, so the total average task allocation rate of the baseline algorithm is 61.57%. Compared with the baseline algorithm, Algorithm 1 improves the task allocation rate by 11.9% because of the use of related task allocation. Compared with the baseline algorithm, the average task allocation rate of Algorithm 2 is 99.34%, which is 38.5% higher than that of the baseline algorithm, which proves the superiority of the algorithm.

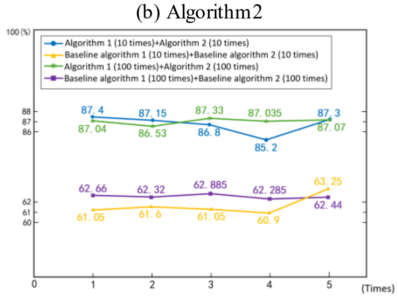
By comparing the first group of data (10 times in each group) with the second group of data (100 times in each group), the fluctuation of the task allocation rate of the algorithm will decrease and the allocation completion rate will tend to be stable with the increase of task execution times.



(a) Algorithm1



(b) Algorithm2



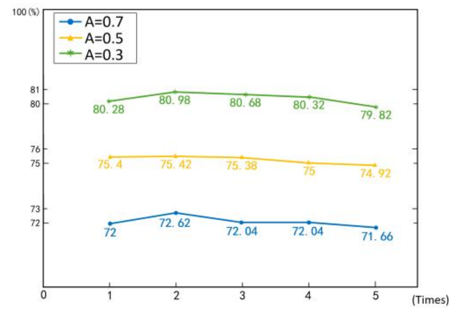
(c) Algorithm1+Algorithm2

Fig. 4. Algorithm task allocation rate result.

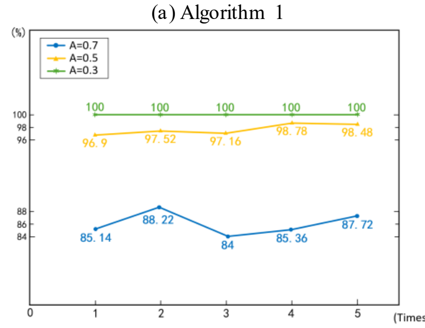
B. Discount Coefficient

We tested three sets of data, and set the discount coefficients a=0.3, a=0.5 and a=0.7, respectively. The urgency ε is randomly generated between 1 and 10, the participants' willingness $PW_{i,j}$ and the correlation coefficient TC（t_i, t_j） are randomly assigned between 0 and 1, the compensation B_i that the task publisher is willing to pay for the task is randomly generated between 1 and 15, and the cost $C_{i,j}$ that the user i needs to perform the task j is randomly generated between 6 and 20. Task set of Algorithm 1 contains 500 randomly generated Tasks, set Part contains 100 randomly generated participants, task set of Algorithm 2 contains 500 randomly generated tasks, and set Part contains 50 randomly generated participants. In the process of testing, each group of data is tested 10 times, and there are five groups. The test results are summarized in Fig. 5.

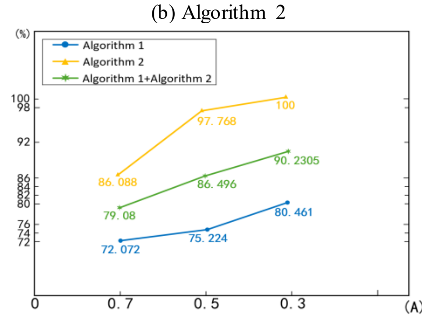
When a=0.7, the average task allocation rate of Algorithm 1 is 72.072%, the average task allocation rate of Algorithm 2 is 86.088%, and the overall average task allocation rate of algorithm is 79.08%. When a=0.5, the average task allocation rate of algorithm 1 is 75.224%, the average task allocation rate of algorithm 2 is 97.768%, and the overall average task allocation rate of algorithm is 86.496%. When a=0.3, the average task allocation rate of Algorithm 1 is 80.416%, the average task allocation rate of Algorithm 2 is 100%, and the total average task allocation rate of algorithm is 90.208%. From the comparison of three data, the smaller the coefficient a is, the greater the cost saved by related allocation, and the greater the incentive provided for the second allocation, the higher the average task allocation rate.



(a) Algorithm 1



(b) Algorithm 2



(c) Algorithm1+Algorithm2

Fig. 5. Under different A values, the task allocation rate.

V. Conclusion

In this study, a related multi-task allocation scheme based on a greedy algorithm is proposed. By quantifying the temporal and spatial distribution characteristics of tasks and participants, different application scenarios are distinguished, and two allocation algorithms suitable for scenarios with sufficient participants and limited resources are designed respectively. Experimental results show that, compared with the benchmark method, the proposed scheme can effectively improve the success rate of task allocation in the mobile crowd sensing (MCS) system in various temporal and spatial distribution scenarios. Especially in scenes with strong task correlation or limited participants' resources, this method shows more stable and consistent performance advantages through joint allocation of related tasks and optimization of incentive costs. Overall, the experimental results verify the effectiveness of the proposed algorithm in improving task allocation efficiency and completion rate from different scene levels.

Although the method in this study has achieved remarkable performance improvement in the experiment, there are still some research limitations. First of all, the task correlation modeling is coarse-grained, which is mainly measured from the perspective of overall correlation, and has not fully described the fine-grained correlation characteristics between users and tasks in terms of historical behavior and ability differences, which limits the applicability of the algorithm in complex practical scenarios to some extent. Secondly, the uncertainties of participants' behaviors, dynamic changes of tasks, communication and calculation overhead in real MCS system have not been fully considered, and the generalization ability of the results needs further evaluation. In addition, although the proposed greedy strategy has advantages in computational efficiency, it still has potential for improvement in global optimality.

Future research will focus on the above limitations. On the one hand, the modeling of task correction and user correction will be further refined, and the multidimensional features and learning mechanism will be introduced to improve the accuracy of the allocation decision. On the other hand, the algorithm will be verified in more complex and real application scenarios, and the multi-task allocation strategy combined with intelligent optimization or learning method will be explored to achieve more efficient and robust MCS multi-task allocation.

References

[1] L. Islam et al., "A blockchain-based crowdsourced task assessment framework using smart contract," International Journal of Advanced Computer Science and Applications, vol. 12, no. 1, pp. 590-600, 2021.

[2] A. S. Alashaikh and F. M. Alhazemi, "Efficient mobile crowdsourcing for environmental noise monitoring," IEEE Access, vol. 10, pp. 77251-77262, 2022.

[3] W. Alasmary, "An innovative smartphone-based solution for traffic rule violation detection," International Journal of Advanced Computer Science and Applications, vol. 11, no. 1, 2020.

[4] Y. Liu, Z. Yu, H. Cui, S. Helal, and B. Guo, "SafeCity: A heterogeneous mobile crowd sensing system for urban public safety," IEEE Internet of Things Journal, vol. 10, no. 20, pp. 18330-18345, 2023.

[5] J. M. Cecilia, J. C. Cano, E. Hernández-Orallo, C. T. Calafate, and P. Manzoni, "Mobile crowdsensing approaches to address the COVID-19 pandemic in Spain," IET Smart Cities, vol. 2, no. 2, pp. 58-63, 2020.

[6] M. Alghasham, M. Alzakan, and M. Al-Hagery, "A review of trending crowdsourcing topics in software engineering highlighting mobile crowdsourcing and AI utilization," International Journal of Advanced Computer Science and Applications, vol. 14, no. 4, 2023.

[7] S. Zhang, Z. Li, W. Liang, K.-C. Li, and Z. A. Bhuiyan, "Blockchain-based hybrid reliable user selection scheme for task allocation in mobile crowd sensing," IEEE Transactions on Network Science and Engineering, 2024.

[8] J. An et al., "Privacy-preserving user recruitment with sensing quality evaluation in mobile crowdsensing," IEEE Transactions on Dependable and Secure Computing, vol. 22, no. 1, pp. 787-803, 2024.

[9] J. Wang, X. Zhan, Y. Yan, and G. Zhao, "A method of user recruitment and adaptation degree improvement via community collaboration in sparse mobile crowdsensing systems," Engineering Applications of Artificial Intelligence, vol. 128, p. 107464, 2024.

[10] G. Yang, D. Guo, B. Wang, X. He, J. Wang, and G. Wang, "Participant-Quantity-Aware Online Task Allocation in Mobile Crowd Sensing," IEEE Internet of Things Journal, 2023.

[11] B. Yin, J. Li, and X. Wei, "Rational task assignment and path planning based on location and task characteristics in mobile crowdsensing," IEEE Transactions on Computational Social Systems, vol. 9, no. 3, pp. 781-793, 2021.

[12] M. M. Rahman et al., "A Framework of Quality-Aware Personalized Task Matching For Mobile Crowdsensing," in 2023 IEEE 13th Symposium on Computer Applications & Industrial Electronics (ISCAIE), 2023: IEEE, pp. 309-314.

[13] J. Ji, Y. Guo, D. Gong, and X. Shen, "Evolutionary multi-task allocation for mobile crowdsensing with limited resource," Swarm and Evolutionary Computation, vol. 63, p. 100872, 2021.

[14] G. Ma, H. Chen, Y. Huang, W. Wei, X. Liu, and Z. Wang, "Utility-based heterogeneous user recruitment of multitask in mobile crowdsensing," IEEE Internet of Things Journal, vol. 10, no. 11, pp. 9796-9808, 2023.

[15] H. Gao, J. Feng, Y. Xiao, B. Zhang, and W. Wang, "A UAV-assisted multi-task allocation method for mobile crowd sensing," IEEE Transactions on Mobile Computing, vol. 22, no. 7, pp. 3790-3804, 2022.

[16] M. Li, M. Ma, L. Wang, B. Yang, T. Wang, and J. Sun, "Multitask-Oriented Collaborative Crowdsensing Based on Reinforcement Learning and Blockchain for Intelligent Transportation System," IEEE Transactions on Industrial Informatics, vol. 19, no. 9, pp. 9503-9514, 2022.

[17] X. Li and X. Zhang, "Multi-task allocation under time constraints in mobile crowdsensing," IEEE Transactions on Mobile Computing, vol. 20, no. 4, pp. 1494-1510, 2019.

[18] Y. Huang et al., "OPAT: Optimized allocation of time-dependent tasks for mobile crowdsensing," IEEE Transactions on Industrial Informatics, vol. 18, no. 4, pp. 2476-2485, 2021.

[19] X. Shen, Q. Chen, H. Pan, L. Song, and Y. Guo, "Variable speed multi-task allocation for mobile crowdsensing based on a multi-objective shuffled frog leaping algorithm," Applied Soft Computing, vol. 127, p. 109330, 2022.

[20] W. Wei, H. Chen, X. Liu, G. Ma, Y. Liu, and X. Liu, "Towards time-constrained task allocation in semi-opportunistic mobile crowdsensing," Ad Hoc Networks, vol. 150, p. 103282, 2023.

[21] M. G. Meitei and N. Marchang, "A Greedy Task Allocation Mechanism for Time-Dependent Mobile Crowdsensing Systems," in 2022 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), 2022: IEEE, pp. 1-6.

[22] J. Wang et al., "Multi-task allocation in mobile crowd sensing with individual task quality assurance," IEEE Transactions on Mobile Computing, vol. 17, no. 9, pp. 2101-2113, 2018.

[23] Y. Liu et al., "Optimizing task allocation with temporal-spatial privacy protection in mobile crowdsensing," Expert Systems, vol. 42, no. 2, p. e13717, 2025.

[24] G. Ye, Y. Zhao, X. Chen, and K. Zheng, "Task allocation with geographic partition in spatial crowdsourcing," in Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 2404-2413.

[25] L. Wang et al., "Towards robust task assignment in mobile crowdsensing systems," IEEE Transactions on Mobile Computing, vol. 22, no. 7, pp. 4297-4313, 2022.

[26] I. Iancu, "Heart disease diagnosis based on mediative fuzzy logic," Artificial intelligence in medicine, vol. 89, pp. 51-60, 2018.

[27] A. Yousefi-Darani, O. Paquet-Durand, and B. Hitzmann, "Application of fuzzy logic control for the dough proofing process," Food and bioproducts processing, vol. 115, pp. 36-46, 2019.

[28] E. Wang, W. Liu, W. Liu, C. Xiang, B. Yang, and Y. Yang, "Spatiotemporal Transformer for Data Inference and Long Prediction in Sparse Mobile CrowdSensing," in IEEE INFOCOM 2023-IEEE Conference on Computer Communications, 2023: IEEE, pp. 1-10.

[29] G. Yang, Y. Zhang, B. Wang, X. He, J. Wang, and M. Liu, "Task allocation through fuzzy logic based participant density analysis in mobile crowd sensing," Peer-to-Peer Networking and Applications, vol. 14, pp. 763-780, 2021.

[30] J. Zhang and X. Zhang, "Multi-task allocation in mobile crowd sensing with mobility prediction," IEEE Transactions on Mobile Computing, vol. 22, no. 2, pp. 1081-1094, 2021.