Participant-Quantity-Aware Online Task Allocation in Mobile Crowd Sensing

Yang, G., Guo, D., Wang, B., He, X., Wang, J. & Wang, G

Original citation & hyperlink:
Yang, G, Guo, D, Wang, B, He, X, Wang, J & Wang, G 2023, 'Participant-Quantity-Aware Online Task Allocation in Mobile Crowd Sensing', IEEE Internet of Things Journal, vol. (In-Press), pp. (In-Press). https://doi.org/10.1109/jiot.2023.3305034

DOI 10.1109/jiot.2023.3305034
ESSN 2327-4662

Publisher: Institute of Electrical and Electronics Engineers

Participant-Quantity-Aware Online Task Allocation in Mobile Crowd Sensing

Guisong Yang, *Member, IEEE*, Dongsheng Guo, Buye Wang, Xingyu He, Jiangtao Wang, *Member, IEEE*, *Gang Wang*

***Abstract*—Task allocation, which can be divided into offline task allocation and online task allocation, is a significant issue in mobile crowd sensing (MCS). Unlike offline task allocation, in the online task allocation scenario, since participants arrive at the service dynamically, the quantity of participants in a specific time and space is uncertain, hence it could affect the quality and efficiency of task completion. However, due to the difficulty of predicting the quantity of real-time participants in a specific time and space accurately, the existing studies of online task allocation lack deep consideration of the quantity of participants. Therefore, this paper investigates a participant-quantity-aware online task allocation problem. First of all, in view of the difficulty of predetermining the participant quantity in MCS, a fuzzy time series analysis (FTSA) method is developed to predict the participant quantity available for each task in a specific time and space. Then, according to the predicted quantity, two reasonable attributes for each task, including the task's threshold on participant's sensing ability and the reward provided for participants to execute the task, can be calculated separately. On this basis, considering the participant's willingness, the participant's sensing ability, the sensor types of the participant's device, and the participant's time coverage jointly, we design an online task allocation algorithm based on an improved genetic algorithm (OTAGA) to allocate appropriate set of tasks to each participant who arrives in real-time, so as to maximize the platform utility and minimize the movement cost of the participant. Simulation results show that the proposed method is effective in terms of the accuracy of prediction, the platform utility and the movement cost of the participant.**

***Index Terms*—Mobile crowd sensing, online task allocation, participant quantity, fuzzy time series analysis, improved genetic algorithm.**

I. Introduction

MOBILE Crowd Sensing (MCS) is an emerging pervasive sensing paradigm that uses a large number of smart mobile terminals (smart phones, tablet computers, sensors, etc.) to collect sensing data with high spatiotemporal correlation and strong analyzability [1]. With the characteristics of wide distribution, high flexibility, and low deployment cost of MCS systems, massive MCS platforms have been established, such as Medusa [2], GigaSight [3], PRISM [4], WAZE [5], and CrowdOS [6]. Under this trend, the MCS provides powerful technical support in intelligent transportation [7], urban public management [8], environmental monitoring [9], and so forth.

A typical MCS system consists of the platform, the task requesters, and the participants, in which the mobile device carried by the participant is regarded as the basic sensing unit [10]. The general process in the MCS system is as follows. Firstly, the task requesters delegate various sensing tasks with detailed task information to the platform. Then, the platform allocates the delegated tasks to appropriate participants. After the tasks are sensed completely, the participants upload the sensing data to the platform. Finally, the platform summarizes and analyzes the sensing data, and feeds back the results to the task requesters.

In the above process, task allocation is regarded as a core issue, which is crucial for the efficiency and effectiveness of MCS applications [11]. Considering different research scenarios, the existing task allocation can be divided into offline task allocation and online task allocation. Offline task allocation generally assumes that participants in the sensing activity are predetermined, and the platform holds their detailed information, such as location, preference, sensor type, etc. Moreover, during the offline task allocation process, it is assumed that neither new participants join the sensing activity in real-time, nor participants suddenly withdraw from the sensing activity. These assumptions are conducive to simplifying the complex task allocation process into a static optimization problem. However, in practical applications, since participants arrive at the service dynamically, it is impossible to obtain accurate information about the participant in advance. Therefore, online task allocation that is more suitable for

This work was supported in part by the National Natural Science Foundation of China under Grants 61802257 and 61602305, in part by the Natural Science Foundation of Shanghai under Grants 18ZR1426000 and 19ZR1477600, in part by the Social Livelihood Planning Project of Nan- tong Science and Technology Bureau under Grant MS12021060, and in part by the Opening Foundation of Agile and Intelligent Computing Key Laboratory of Sichuan Province. *(Corresponding author: Xingyu He.)*

G. Yang is with the Department of Computer Science and Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China, and also with the Agile and Intelligent Computing Key Laboratory of Si-chuan Province, Chengdu 610000, China. (e-mail: gsyang@usst.edu.cn).

D. Guo is with the Department of Computer Science and Engineering at University of Shanghai for Science and Technology, Shanghai 200093, China (e-mail: 202580527@st.usst.edu.cn).

B. Wang is with the Department of Computer Science and Engineering at University of Shanghai for Science and Technology, Shanghai 200093, China (e-mail: 182590511@st.usst.edu.cn).

X. He is with College of Communication and Art Design at University of Shanghai for Science and Technology, Shanghai 200093, China (e-mail: xy_he@usst.edu.cn).

J. Wang is with the Center for Intelligent Healthcare, Coventry University, Coventry CV1 5RW, United Kingdom (email: jiangtao.wang@coventry.ac.uk).

G. Wang is with MaxLinear Inc., California, 92618, United States (e-mail: gwang.buaa@gmail.com)

practical applications has gradually become a research hotspot and this paper focuses on the online task allocation for MCS.

In an online task allocation scenario, MCS tasks generally have spatiotemporal requirements (e.g., if the platform wants to monitor the air quality of a park from 18:00 to 20:00, it needs to recruit participants who arrive at the park during this time period. Participants who arrive at other times will not be able to provide any help.) and sensor requirements (If a participant is asked to collect urban noise data, but there is no sound sensor on his mobile device, so he does not meet the requirements of the task). For a task, the participant quantity available for it in a specific time and space has a great influence on whether the task can be well completed, where the availability of participants depends on a variety of factors including the sensor types of participants' devices. For example, insufficient available participants may lead to a long task completion cycle. On the contrary, when the participant quantity available for a task is sufficient, the platform can recruit participants with strong sensing ability to improve the quality of task completion. Therefore, for an online task allocation, the quality and efficiency of task completion depend on the participant quantity available for a task in a specific time and space.

However, since the participants usually arrive at the service dynamically, it is difficult for the platform to predetermine the participant quantity available for a task in a specific time and space. In this case, in order to facilitate the platform in formulating a reasonable online task allocation scheme effectively, it is essential to accurately predict the participant quantity available for each task in a specific time and space. Based on the above discussion, to achieve optimal online task allocation, the first challenge is how to predict the participant quantity available for each task in a specific time and space accurately. The second challenge is how to formulate a reasonable online task allocation scheme based on the predicted participant quantity.

For the first challenge, there are two main factors that affect the prediction of the participant quantity available for each task in a specific time and space. On one hand, since participants can join in and withdraw from the sensing activities flexibly, their sensing range and availability at different time are variable and uncertain, which causes the participant quantity available for a task varies and fluctuates greatly with time. On the other hand, there are many uncertainties (such as GPS signal interruption, missing trajectory data, etc.) during the process of participating in sensing activities, which leads to the historical data on the participant quantity available for a task in a specific time and space is incomplete and ambiguous. Due to the above two factors, the traditional prediction methods with high requirements for the integrity and accuracy of historical data are not suitable for solving the problem of predicting the participant quantity in this paper. In order to solve the first challenge, a fuzzy time series analysis (FTSA) method is developed to predict the participant quantity available for each task in a specific time and space. The FTSA method refers to use fuzzy mathematics method to study the characteristics and development trends contained in data sequences with fuzzy and incomplete information. Since the participant quantity varies with time, the change of the fuzzy language used to describe participant quantity is also dynamically dependent on time series. In addition, due to incomplete and ambiguous historical data of participant quantity, it is more realistic to use fuzzy language such as less, normal, and more to describe the participant quantity in practical applications.

To solve the second challenge, firstly, we set two reasonable attributes for each task, including the task's threshold on the participants' sensing ability and a reasonable reward provided for participants to execute the task, and then estimate them based on the predicted participant quantity. Secondly, in the process of online task allocation, this paper considers two optimization objectives from the perspective of the platform and the participants, which are the platform utility and the movement cost of the participant. Maximizing the platform utility can ensure that the platform obtains high-quality sensing data, and minimizing the movement cost of the participant can ensure that the participant has the lowest movement cost during the time period he perform tasks. Finally, considering the participant's willingness, the participant's sensing ability, the sensor types of the participant's device, and the participant's time coverage jointly, we design an online task allocation algorithm based on an improved genetic algorithm to allocate appropriate tasks to the arrived participants to achieve the above two optimization objectives.

Since the above online task allocation problem is a bi-objective optimization problem, the designed algorithm is different from the traditional single-objective optimization genetic algorithm. Firstly, unlike most genetic algorithms using binary coding structures or fixed-length coding structures to construct chromosomes, the designed algorithm adopts a variable-length symbol coding structure to construct chromosomes, where a chromosome represents a task allocation scheme, and in each chromosome the participant and tasks are regarded as genes. The order of the genes represents the order in which the participants perform the tasks, and the participant is located at the beginning of each chromosome. Secondly, the designed algorithm adopts lexicographical order to indicate the fitness of chromosomes in the population, and determines the probability of chromosomes being selected for the next generation through a ranking-based selection method. Chromosomes ranked higher represent that the task allocation schemes can provide the platform with higher utility and make the movement cost of the participant less. Then, the designed algorithm uses interchromosomal crossover and intrachromosomal crossover to expand the solution space of the two optimization objectives respectively. Interchromosomal crossover swaps different tasks in two different task allocation schemes to seek higher platform utility, and intrachromosomal crossover changes the task execution order of the participant in a task allocation scheme to make the movement cost of the participant less. Finally, the algorithm uses mutation operations to change tasks in one task allocation scheme into other tasks that do not exist in that scheme to avoid falling into local optimality.

The main contributions are summarized as follows:

(1) We investigate the significant impact of participant quantity on the quality and efficiency of task completion, and analyze the difficulty of predetermining the participant quantity in MCS. Then, a FTSA method is developed to predict the participant quantity available

for each task in a specific time and space.

(2) According to the predicted participant quantity, the task's threshold on participants' sensing ability and the reward provided for participants to execute the task can be calculated separately. Then, considering multiple attributes of tasks and participants comprehensively, we design an online task allocation algorithm based on an improved genetic algorithm to allocate appropriate tasks to each participant who arrives dynamically, so as to maximize the platform utility and minimize the movement cost of the participant.

The rest of the paper is organized as follows. In Section II, the related work of offline task allocation and online task allocation are introduced. In Section III, an online task allocation model is introduced. Section IV designs an online task allocation algorithm. The performance evaluation and discussion are given in Section V. Section VI concludes the paper.

II. Related Work

In this section, the recent state of the art on offline task allocation and online task allocation is presented.

A. *Offline Task Allocation in MCS*

Since there are many factors that affect the MCS system in practical applications, it is an effective way to abstract the task allocation process into a static model. Early research works on MCS focused on an offline task allocation model with predetermined information of tasks and participants.

In [12], a duration-sensitive task allocation model, where each task is associated with a specific sensing duration, was designed to solve the offline task allocation problem with task duration. Similarly, the authors of [13] studied a multi-task allocation problem, which investigated the impact of time constraints and maximized the platform utility. Considering the stability of task allocation, Dai et al. [14] constructed a distributed many-to-many matching model to capture the interaction between task requesters and workers. The authors of [15] proposed a stable task allocation algorithm, which achieved a stable MCS system through the stable marriage approach. Considering the privacy protection in the MCS system, Zhang et al. [16] proposed a novel differentially private geocoding (DPG) mechanism to preserve workers' location privacy. In [17], an incentive mechanism was designed to prevent malicious workers from exploring illegal benefits by simply uploading falsified parameters. Considering the important influence of participants' willingness on sensing data quality, the authors of [18] proposed a willingness and trust-based collaborative team recruitment method (WT-CTRM) to improve the quality of service (QoS) among recruited users. Similarly, the authors of [19] proposed a QoI-aware energy-efficient participant selection approach to collect sufficient amount of sensory data with high quality-of-information (QoI) requirements. In [20], Wu et al., established a regression model based on user willingness using a fully connected deep neural network to quantitatively evaluate the user's willingness to perform the perceptual task.

The aforementioned researches consider many factors in the offline task allocation scenario, and assume that the information of tasks and participants is available in advance. In contrast, in this paper, task allocation is regarded as a dynamic process, and tasks need to be allocated to participants who arrive at the service dynamically.

B. *Online Task Allocation in MCS*

With the development of MCS, the online task allocation has gradually become a research hotspot, where the platform allocates tasks to participants who arrive at the service dynamically.

In [21], aiming at the problem of quality-aware online task assignment, a probabilistic model to measure the quality of tasks and a hitchhiking model to characterize workers' behavior patterns were proposed. The authors of [22] studied a dynamic participant selection problem with heterogeneous sensing tasks, which minimized the sensing cost while maintaining a specific level of probabilistic coverage. In [23], a semi-Markov model was proposed to predict the position distribution of workers and tasks, and a prediction-based task allocation algorithm (PBTA), which could gain the maximum global system utility and lowest traveling cost, was designed to solve a dynamic task allocation problem. In [24], Li et al. predicted the mobility model of opportunistic workers and proposed a Multi-stage multi-task Online Task Assignment (MOTA) algorithm to minimize the average make span of all tasks. The authors of [25] proposed a hybrid framework named HyTasker which can allocate tasks to opportunistic workers in the offline phase and participatory workers in the online phase to realize the complementary advantages of the two modes. The authors of [26] proposed a task allocation framework Re-OPSEC, which can effectively select opportunistic workers at real time to perform tasks in an energy-saving way. Wang et al. [27] studied the location-aware and location diversity-based dynamic crowd sensing system, and proposed an online control policy to ensure the fairness of workers. Similarly, the authors of [28] designed a polynomial-time approximation algorithm to allocate each of the sequentially arriving tasks to participants as fairly as possible. In [29], an online multi-task allocation mechanism with a defined reliability metric, which was associated with a rational payment model, was proposed to achieve the online task allocation.

In contrast to the above studies, in this paper, considering that the quantity of participants has an important impact on the quality and efficiency of task completion, we therefore analyze it in detail. In response to this intractable problem, the FTSA method is designed to predict the participant quantity available for each task in a specific time and space, and then an online task allocation algorithm based on an improved genetic algorithm is designed to allocate appropriate tasks to each participant.

III. Online Task Allocation Model

In this paper, we consider an online task allocation scenario, where participants arrive at the service dynamically, and both participants and tasks have multiple attributes. Different from the assumption that all participants are predetermined in offline task allocation, the platform can only obtain the information of the participants who arrive at the service in online task allocation. According to the arrived participants' information, the platform allocates an appropriate set of tasks to them.

The system model of this paper can be described as follows. Firstly, the task requester submits task information to the platform. Then, the platform adopts a FTSA method to predict the participant quantity available for the task in a specific time and space. After, according to the predicted participant quantity, the task's threshold on participants' sensing ability and the reward provided for participants to execute the task are calculated separately. Afterwards, when a participant arrives at the service dynamically, the platform allocates appropriate tasks to the participant according to the attributes of both the participant and tasks. Finally, when the task is completed, the participant needs to upload the sensing data to the platform, and the platform returns the aggregated and processed data to the task requester.

To indicate the aforementioned system model clearly, the detailed parameters used are introduced as follows. In this paper, a day is divided into z time periods with equal length $C_Z = \{C_1, \ldots, C_d, \ldots, C_z\}$, and the sensing area is divided into l independent sub-areas $S_L = \{S_1, \ldots, S_e, \ldots, S_l\}$. In order to facilitate the subsequent description, a sub-area S_e within a time period C_d is defined as a spatiotemporal granularity $G_{d,e}$ (i.e., a spatiotemporal unit [30]). Within a spatiotemporal granularity $G_{d,e}$, the set of n tasks is denoted as $T_N = \{T_1, \ldots, T_i, \ldots, T_n\}$, and the participant quantity that task T_i requires is denoted as $M_{d,e}^{i}$. As shown in Fig.1, a sensing area is divided into 12 independent sub-areas, and the span of each time period is 2 hours. Different tasks are represented by symbols with different shapes and the color of the symbol represents the degree of the participant quantity available for each task. The number next to the symbol represents the predicted participant quantity available for each task in its spatiotemporal granularity. In order to formulate a suitable online task allocation scheme, it is necessary to predict the participant quantity available for each task in its spatiotemporal granularity.

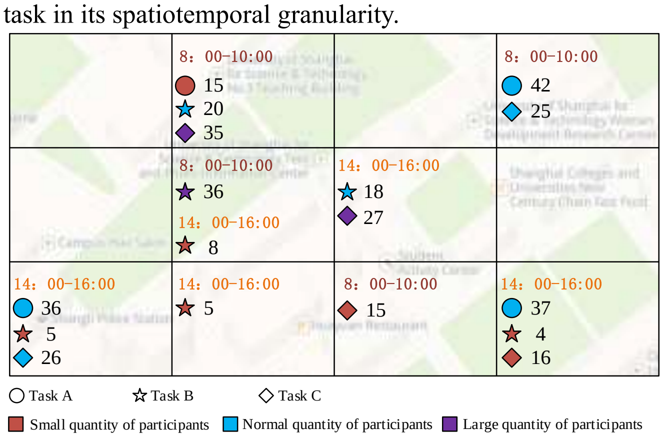

Fig.1. The predicted participant quantity in time and space.

A. Participant quantity Prediction

In MCS, the available participant quantity will affect the quality and efficiency of task completion. However, in online task allocation, since participants arrive at the service dynamically and their information is unknown, the platform cannot determine the available participant quantity for tasks in advance. Therefore, this paper adopts the FTSA method to predict the participant quantity available for each task in its spatiotemporal granularity. For the convenience of understanding, the theoretical foundations are defined as follows:

Definition 1. Let U be the universe of discourse, where $U = \{u_1, u_2, \ldots, u_n\}$. A fuzzy set FS_i on U is denoted as

$$FS_i = \frac{f_{FS_i}(u_1)}{u_1} + \frac{f_{FS_i}(u_2)}{u_2} + \cdots + \frac{f_{FS_i}(u_n)}{u_n}, \tag{1}$$

where f_{FS_i} is the fuzzy membership function defined on the fuzzy set FS_i. $f_{FS_i}(u_k)$ represents the fuzzy membership of u_k to fuzzy set FS_i, and $f_{FS_i}(u_k) \in [0, 1]$, $1 \le k \le n$.

Definition 2. A group of fuzzy sets $f_i(t)$ ($i = 1, 2, \ldots$) is defined on a subset of real numbers $Y(t) \subset \mathbf{R}$ ($t = \ldots, 0, 1, 2, \ldots$). $F(t) = \{f_1(t), f_2(t), \ldots\}$ is defined as a fuzzy time series on $Y(t)$.

Definition 3. Let $F(t-1) = FS_i$ and $F(t) = FS_j$ the fuzzy logical relation between $F(t\text{-}1)$ and $F(t)$ can be denoted by $FS_i \rightarrow FS_j$.

As shown in Fig.2, the steps of the FTSA method are introduced as follows. i) Fuzzification. Fuzzy the training set, divide the universe of discourse, and determine the fuzzy set and fuzzy membership function. ii) Establishment of fuzzy logical relation set. Establish the fuzzy logical relation set of the sample data according to the sequence of the training data. iii) Establishment of fuzzy logical relation matrix. Establish the fuzzy logical relation matrix according to all fuzzy logical relations. iv) Prediction. Obtain the predicted value according to the fuzzy logical relation matrix and the given prediction rules.

Fuzzification. The FTSA method needs to determine the discussion scope of the problem (i.e., the universe of discourse) according to the minimum and maximum values of historical data. For the convenience of discussion and calculation, the universe of discourse is usually rounded down and up to get the lower boundary Q_{min} and the upper boundary Q_{max}. Then, the universe of discourse is divided into several fuzzy subintervals, that is, the universe of discourse is divided in a way that can be understood as much as possible by natural language according to the actual situation. In this paper, the universe of discourse is evenly divided into seven fuzzy subintervals $\{u_1, u_2, \ldots u_7\}$. The semantics are "FS_1: few participants", "FS_2: a few participants", "FS_3: small participant quantity", "FS_4: normal participant quantity", "FS_5: large participant quantity", "FS_6: a lot of participants", "FS_7: massive participants".

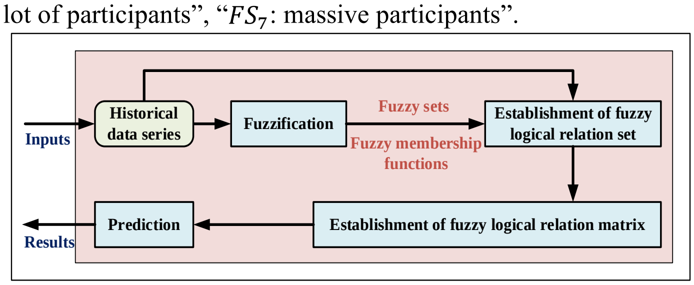

Fig.2. Steps of fuzzy time series model.

TABLE I
FUZZY SETS AND FUZZY MEMBERSHIP VALUES

Days	Observed Value	FS_1	FS_2	FS_3	FS_4	FS_5	FS_6	FS_7	Fuzzification
day 1	12	0.0833	0.5833	0.9167	0.4167	0	0	0	FS_3
day 2	13	0	0.4167	0.9167	0.5833	0.0833	0	0	FS_3
day 3	12	0.0833	0.5833	0.9167	0.4167	0	0	0	FS_3
day 4	5	1	0.25	0	0	0	0	0	FS_1
day 5	13	0	0.4167	0.9167	0.5833	0.0833	0	0	FS_3
day 6	16	0	0	0.4167	0.9167	0.5833	0.0833	0	FS_4
day 7	7	0.9167	0.5833	0.0833	0	0	0	0	FS_1
day 8	7	0.9167	0.5833	0.0833	0	0	0	0	FS_1
day 9	12	0.0833	0.5833	0.9167	0.4167	0	0	0	FS_3
day 10	26	0	0	0	0	0	0.25	1	FS_7
day 11	7	0.9167	0.5833	0.0833	0	0	0	0	FS_1
day 12	10	0.4167	0.9167	0.5833	0.0833	0	0	0	FS_2

The fuzzy sets are defined by the triangular fuzzy membership function as

$$\begin{cases} FS_1 = \frac{1}{u_1} + \frac{0.5}{u_2} + \frac{0}{u_3} + \frac{0}{u_4} + \frac{0}{u_5} + \frac{0}{u_6} + \frac{0}{u_7} \\ FS_2 = \frac{0.5}{u_1} + \frac{1}{u_2} + \frac{0.5}{u_3} + \frac{0}{u_4} + \frac{0}{u_5} + \frac{0}{u_6} + \frac{0}{u_7} \\ FS_3 = \frac{0}{u_1} + \frac{0.5}{u_2} + \frac{1}{u_3} + \frac{0.5}{u_4} + \frac{0}{u_5} + \frac{0}{u_6} + \frac{0}{u_7} \\ FS_4 = \frac{0}{u_1} + \frac{0}{u_2} + \frac{0.5}{u_3} + \frac{1}{u_4} + \frac{0.5}{u_5} + \frac{0}{u_6} + \frac{0}{u_7}. \\ FS_5 = \frac{0}{u_1} + \frac{0}{u_2} + \frac{0}{u_3} + \frac{0.5}{u_4} + \frac{1}{u_5} + \frac{0.5}{u_6} + \frac{0}{u_7} \\ FS_6 = \frac{0}{u_1} + \frac{0}{u_2} + \frac{0}{u_3} + \frac{0}{u_4} + \frac{0.5}{u_5} + \frac{1}{u_6} + \frac{0.5}{u_7} \\ FS_7 = \frac{0}{u_1} + \frac{0}{u_2} + \frac{0}{u_3} + \frac{0}{u_4} + \frac{0}{u_5} + \frac{0.5}{u_6} + \frac{1}{u_7} \end{cases} \quad (2)$$

The membership function of each fuzzy set FS_i is expressed as

$$u_{FS_i}(t) = \begin{cases} 1, \ if \ i = 1 \ and \ x_t \le m_1 \\ 1, \ if \ i = 7 \ and \ x_t \ge m_7 \\ max\left\{0, 1 - \frac{|x_t - m_i|}{2 \times l}\right\}, \ other \end{cases}, \quad (3)$$

where l represents the length of the subinterval, t represents the time, x_t represents the observed value at time t, and m_i represents the central value of the subinterval u_i.

Establishment of fuzzy logical relation set. According to Eq. (3), the fuzzy membership of the observed value to each fuzzy set can be calculated, and then the fuzzy set to which the observed value belongs can be determined according to its maximum fuzzy membership. After calculating the fuzzy sets to which all observed values in the training set belong, according to definition 3, the fuzzy logical relation $FS_i \rightarrow FS_j$ between two adjacent observed values can be obtained. Finally, a set containing all fuzzy logical relations can be established.

Establishment of fuzzy logical relation matrix. By counting the occurrence times of all fuzzy logical relations, the fuzzy logical relation matrix can be established as

$$\boldsymbol{RM} = \left\{ \begin{matrix} n_{11} & \cdots & n_{1j} & \cdots & n_{17} \\ \vdots & & \vdots & & \vdots \\ n_{i1} & \cdots & n_{ij} & \cdots & n_{i7} \\ \vdots & & \vdots & & \vdots \\ n_{71} & \cdots & n_{7j} & \cdots & n_{77} \end{matrix} \right., \quad (4)$$

where n_{ij} represents the occurrence times of the fuzzy logical relation $FS_i \rightarrow FS_j$ in all fuzzy logical relations.

Prediction. The prediction value can be obtained by calculating the weighted average of the elements in the fuzzy logical relation matrix to the central value of the fuzzy sets. The prediction rule is expressed as

$$Fval(t+1) = \begin{cases} m_{c_t}, \ \sum_{j=1}^{n} \boldsymbol{RM}(c_t, j) = 0 \\ \frac{\boldsymbol{RM}(c_t,:)}{\sum_{j=1}^{n} \boldsymbol{RM}(c_t, j)} \times (m_1, m_2, \ldots, m_n)^T, \ other \end{cases}, \quad (5)$$

where $Fval(t+1)$ is the final predicted value. m_{c_t} is the central value of the subinterval corresponding to the fuzzy set to which the observed value belongs. $\boldsymbol{RM}(ct,:)$ is the row vector in the fuzzy logical relation matrix corresponding to the fuzzy set to which the observed value belongs. $(m_1, m_2, \ldots, m_n)^T$ is the transpose of $(m_1, m_2, \ldots, m_n)$, where m_i is the central value of the subinterval u_i. The symbol "×" means vector multiplication. Through Eq. (5), the predicted participant quantity available for each task in its spatiotemporal granularity can be obtained.

An instance of the FTSA method is depicted in Table I. We provide the observed participant quantity available for a task in its spatiotemporal granularity within 12 days, and predict the available participant quantity on the day 12 based on the observed value of 11 days. In this instance, the universe of discourse is [5,26], which can be divided into seven subintervals {[5, 8], [8, 11], [11, 14], [14, 17], [17, 20], [20, 23], [23, 26]}. According to the fuzzy membership values in Table I, the fuzzification results of each day can be obtained. Thus, all fuzzy logical relations of 11 days can be obtained, i.e., $FS_3 \rightarrow FS_3$, $FS_3 \rightarrow FS_3$, $FS_3 \rightarrow FS_1$, $FS_1 \rightarrow FS_3$, $FS_3 \rightarrow FS_4$, $FS_4 \rightarrow FS_1$, $FS_1 \rightarrow FS_1$, $FS_1 \rightarrow FS_3$, $FS_3 \rightarrow FS_7$, $FS_7 \rightarrow FS_1$. Then, the fuzzy logical relation matrix can be established as

$$\boldsymbol{RM} = \left\{ \begin{matrix} 1 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 2 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0. \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \right.$$

Finally, the predicted value can be calculated as $Fval(day\ 12)$ =1/3 * [1, 0, 2, 0, 0, 0, 0] × [6.5, 9.5, 12.5, 15.5, 18.5, 21.5, 24.5] = 10.5, which is similar to the observed value on the day 12.

B. Model Foundation

In the online task allocation scenario, the arrived participants select tasks that they are willing to execute according to the tasks' information. The platform allocates a set of tasks to each participant based on the participant's willingness, the participant's sensing ability, the sensor types of the participant's device, and the participant's time coverage. For example, as shown in Fig.3, it is assumed that there are two tasks T_1 and T_2 need to be allocated within 8:00 to 10:00, and three participants arrive at the service within this time period. In addition, assuming that it takes 40 and 20 minutes to complete task T_1 and task T_2 and the three participants P_1, P_2, and P_3 arrive at the service at 8:00, 8:30, and 9:30 respectively. For task T_1, participants P_1, P_2, and P_3 all select it, where the sensing ability and sensor types of participants P_1 and P_3 meet T_1's requirements, and participant P_2 does not meet these two requirements. However, since the remaining activity time of participant P_3 can no longer meet the time requirement for executing task T_1, the platform only allocates task T_1 to participant P_1 in the end. For task T_2, only participant P_3 selects it, and participant P_3 satisfies all requirements of task T_2, so finally the platform allocates task T_2 to participant P_3.

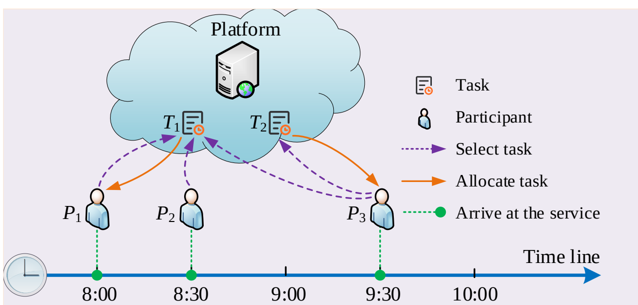

Fig.3. Online scenario.

According to the predicted participant quantity, the platform can calculate the task's threshold on participants' sensing ability and the reward provided for participants to execute the task. The participants' sensing ability will affect the quality of task completion. In order to ensure the quality of task T_i, the task requester can set a threshold of $0 \le \varepsilon_{low}^i \le 1$ on the participants' sensing ability, which is the minimum sensing ability the participant required to execute the task. When the predicted participant quantity available for the task is small, a relatively low threshold can be set to recruit more participants for the task to meet the platform's requirements for the quantity of participants. On the contrary, when the predicted participant quantity available for the task is large, a relatively high threshold can be set to recruit participants with strong sensing ability to improve the quality of the task completion. According to the predicted quantity $N_{d,e}^i$ of participants available for task T_i in spatiotemporal granularity $G_{d,e}$, the threshold of task T_i on the participants' sensing ability can be calculated as

$$\varepsilon_i = \begin{cases} (\varepsilon_b^i - \varepsilon_{low}^i) * \frac{N_{d,e}^i}{M_{d,e}^i} + \varepsilon_{low}^i, N_{d,e}^i \le M_{d,e}^i \\ (1 - \varepsilon_b^i) * \frac{N_{d,e}^i - M_{d,e}^i}{N_{d,e}^i} + \varepsilon_b^i, other \end{cases}, \quad (6)$$

where $M_{d,e}^i$ represents the participant quantity the task T_i requires, which is an attribute of the task and is determined by the task requester. ε_b^i represents a benchmark threshold of task T_i, which is set to ensure that the participant quantity whose sensing ability is not less than the threshold is equal to $M_{d,e}^i$. ε_b^i satisfies the condition $0 \le \varepsilon_{low}^i \le \varepsilon_b^i \le 1$ and can be calculated by training historical data.

After participants complete tasks, the platform needs to provide rewards for them. When the predicted participant quantity available for the task is small, the platform can increase the reward of the task to encourage more participants to perform the task. On the contrary, when the predicted participant quantity available for the task is large, the platform can reduce the reward of the task to save costs. Different from the task's threshold on participants' sensing ability, the change of reward provided for participants to execute the task in practical application is generally coarse-grained, so the reward of task T_i is designed as a piecewise function based on the divided fuzzy subintervals $\{u_1, u_2, \dots u_7\}$ in Section III-A, which is calculated as

$$r_i = \begin{cases} r_{high}^i - \frac{sub-1}{3} * (r_{high}^i - init\_r_i), & sub \le 4 \\ r_{low}^i + \frac{7-sub}{3} * (init\_r_i - r_{low}^i), & other \end{cases}, \quad (7)$$

where *sub* represents that the predicted participant quantity $N_{d,e}^i$ is distributed in the *sub*-th fuzzy subinterval. $init\_r_i$ represents the initial reward of task T_i. r_{low}^i and r_{high}^i represent the lowest and highest reward of task T_i respectively, which can be calculated by training historical data.

In this paper, two optimization objectives from the perspective of the platform and the participants are considered, which are the platform utility and the movement cost of the participant. Maximizing the platform utility can ensure that the platform obtains high-quality sensing data, and minimizing the movement cost of the participant can ensure that the participant has the lowest movement cost during the time period he perform tasks.

The platform utility. The platform utility is relevant to many factors, such as the participant's willingness, the participant's sensing ability, the sensor types of the participant's device, and the participant's time coverage.

Firstly, the participant's willingness to perform tasks is an important factor affecting the quality of sensing data. The participant's willingness to the task reflects his/her preference for the task. The participant P_j's willingness to task T_i is denoted as $h_{i,j}$, where $0 \le h_{i,j} \le 1$. In addition, the participant P_j's threshold on the expected reward of task T_i is denoted as $g_{i,j}$. A participant will not select a task whose reward is lower than its expectation, i.e., when $r_i < g_{i,j}$, $h_{i,j} = 0$.

Secondly, the participant's sensing ability to execute tasks is different. The participant P_j's sensing ability to execute task T_i is denoted as $\varepsilon_{i,j}$, where $0 \le \varepsilon_{i,j} \le 1$. Only if participant P_j's sensing ability meets task T_i's threshold on the participants' sensing ability, i.e., $\varepsilon_{i,j} \geqslant \varepsilon_i$, the task can be allocated to the participant.

Thirdly, a sensing task may require one or more types of sensors to collect the needed information, and the devices of different participants are equipped with different sensors. Therefore, when the platform allocates tasks to a participant, it needs to consider whether the sensors in the participant's device meet the sensor requirements of the task. If the sensors in the participant P_j's device meet the sensor requirements of task T_i, $w_{i,j}$ = 1; otherwise, $w_{i,j}$ = 0.

Finally, since the participant arrives at the service dynamically, the platform also needs to consider the participant's time coverage. The remaining activity time of participant P_j is denoted as $Time_j$, the movement time that participant P_j takes to move to the location of task T_i is denoted as $Time^m_{i,j}$, and the sensing time participant P_j takes to execute task T_i is denoted as $Time^s_{i,j}$. For the convenience of the following description, a set of c tasks that participant P_j is willing to execute and meets the tasks' sensor requirements is called P_j's candidate set, which is denoted as T^j_C. A set of k tasks allocated to participant P_j is called P_j's allocation set, which is denoted as T^j_A. Whether participant P_j meets the time coverage requirement of allocation set T^j_A is denoted as $\delta_j(T^j_A)$, where $\delta_j(T^j_A)$ = 1 means participant P_j meets the time coverage requirement of T^j_A; otherwise, $\delta_j(T^j_A) = 0$. The condition of $\delta_j(T^j_A)$ = 1 can be formulated as

$$Time_j \geq \sum_{i=1}^{k}(Time^m_{i,j} + Time^s_{i,j}). \quad (8)$$

According to the aforementioned factors, the utility that participant P_j brings to the platform can be formulated as

$$Utility_j = \sum_{i=1}^{k} h_{i,j} * \varepsilon_{i,j} * w_{i,j} * \delta_j(T^j_A). \quad (9)$$

Movement cost. To calculate the movement cost of the participant, we consider the following factors that affect the movement cost of the participant. Firstly, the unit movement cost of participants refers to the cost of participants walking a unit distance in the process of executing tasks. In this paper, the movement cost per unit distance of participant p_j is denoted as uc_j. Moreover, when a participant walks a unit distance, its intelligent device will consume corresponding resources, such as electricity, cell phone data, etc. Therefore, in this paper, we define the unit consumption cost of participant p_j's intelligent device as dc_j. Finally, the total movement distance of the participant in the process of performing tasks is a crucial factor affecting the movement cost of the participant. The distance between two locations is calculated as

$$Distance_{p,q} = |x_p - x_q| + |y_p - y_q|, \quad (10)$$

where x_p and x_q represent the abscissa of the two locations, y_p and y_q represent the ordinate of the two locations.

The total distance that participant P_j moves to execute all tasks in its allocation set T^j_A with k tasks can be calculated as

$$Distance_j = Distance_{P_j,T_1} + \sum_{i=1}^{k-1} Distance_{T_i,T_{i+1}}. \quad (11)$$

The total movement cost of participant P_j can be calculated as

$$Cost_j = uc_j * dc_j * Distance_j. \quad (12)$$

In this paper, we study the participant-quantity-aware online task allocation problem with the objectives of maximizing the platform utility and minimizing the movement cost of the participant. The problem can be formulated as

$$\text{maximize } Utility_j, \quad (13)$$

$$\text{minimize } Cost_j. \quad (14)$$

Subject to:

$$0 \leq h_{i,j} \leq 1, \quad (15)$$

$$g_{i,j} \leq r_i, \quad (16)$$

$$\varepsilon_{i,j} \geq \varepsilon_i, \quad (17)$$

$$w_{i,j} \in \{0,1\}, \quad (18)$$

$$\delta_j(T^j_A) \in \{0,1\}. \quad (19)$$

IV. Online Task Allocation Algorithm

The above bi-objective optimization problem has a huge solution space, which may not be solved by some algorithms in polynomial time. Genetic algorithm is used to search for the optimal solution by simulating natural evolution phenomena, which has been proved to be an effective combinatorial optimization algorithm [31]. Therefore, in this section, we design an online task allocation algorithm based on an improved genetic algorithm to solve the bi-objective optimization problem. In the designed algorithm, the participant and tasks are regarded as the genes in the chromosome. Through the algorithm, a chromosome that can maximize the platform utility and minimize the movement cost of the participant can be obtained, which can also determine the task execution route of the participant. The designed algorithm includes population initialization, chromosome representation, fitness function, selection operation, crossover operation, and mutation operation, which are described in detail as follows.

Population initialization. In this paper, a task allocation scheme includes a task set allocated to the participant and the task execution route of the participant. The purpose of population initialization is to generate multiple feasible solutions in the solution space, that is, to generate multiple feasible task allocation schemes in MCS. In order to make the generated feasible solutions uniformly distributed in the solution space, we initialize the population randomly. For participant P_j, the designed algorithm randomly selects several tasks from its candidate set T^j_C and randomly determines its task execution route, and repeats this process until an initial population with N chromosomes is formed.

Chromosome representation. In this paper, a chromosome represents a task allocation scheme in MCS. Genetic algorithms mainly use binary coding structures and fixed-length coding structures in most systems, but these coding structures are not suitable for solving the online task allocation problem in this paper. The reason is that, on the one hand, the binary coding structure cannot represent different task execution routes. On the other hand, since the tasks allocated to the participant is a variable subset of the participant's candidate set, the fixed-length coding structure is not suitable. Therefore, we designed the chromosome representation as shown in Fig.4. Specifically, there are N chromosomes, and each chromosome contains at most c + 1 genes, where c is the number of tasks in the candidate set of the participant. The first gene of each chromosome represents the participant, and others represent different tasks. For example, in Fig.4, chromosome 1 represents that the allocation set of participant P_j is $\{T_2, T_1, T_3\}$, and its task execution route is $T_2 \rightarrow T_1 \rightarrow T_3$.

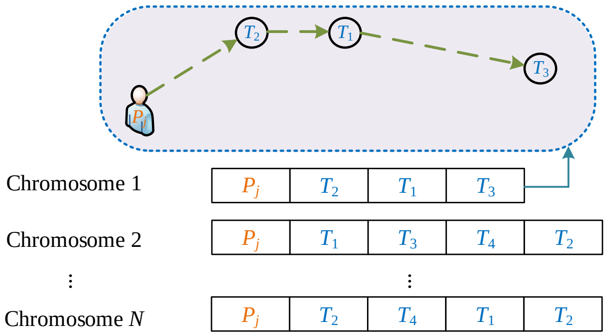

Fig.4. Chromosome representation.

Fitness function. In the designed algorithm, the fitness function is the main indicator describing the performance of the task allocation scheme. In the optimization problem, the mapping relationship between the optimization goal and chromosomal fitness can be established, and the optimization goal can be achieved through the survival of the fittest in the evolutionary process. In this paper, the main optimization goal is to maximize the platform utility, based on which, another optimization goal is to minimize the movement cost of the participant. Therefore, the platform utility should be the main factor affecting fitness. The algorithm prioritize maximizing the platform utility, and it then further optimize the movement cost of the participant. Based on the above analysis, the fitness is designed in the form of lexicographic order [32]. The concept of lexicographical order is defined as follows.

Definition 4. For two totally ordered sets A and B, (a, b) and (a', b') belong to the Cartesian Product $A \times B$, then the lexicographic order is defined as

$$(a, b) \leq (a', b'), if\ a < a'\ or\ (a = a'\ and\ b \leq b'). \quad (20)$$

Since the fitness is directly proportional to the platform utility and inversely proportional to movement cost, the fitness function can be expressed as

$$fitness = (Utility_j, \frac{1}{Cost_j}). \quad (21)$$

Selection operation. The selection operation is to select chromosomes for the generation of new chromosomes. In this paper, since the fitness is designed in lexicographic order, a ranking-based selection method is designed to perform the selection operation as follows. i) Sort chromosomes in ascending order according to their fitness. ii) Calculate the probability of each chromosome being selected according to its ranking. iii) Select chromosomes based on the probability. The probability of the i-th chromosome being selected can be calculated as

$$P_i = P_{min} + (P_{max} - P_{min}) * \frac{i-1}{N-1}, \quad (22)$$

where P_{max} and P_{min} represent the probability of the best and worst chromosomes being selected, respectively.

Crossover operation. A batch of chromosomes can be obtained by a selection operation, and then a crossover operation is performed to obtain more chromosomes. To achieve the two optimization goals, a two-step crossover operation is designed. The first step is the interchromosomal crossover operation, which can obtain a larger solution space for the platform utility maximization by adjusting the allocation set. The interchromosomal crossover includes single-point crossover and multi-point crossover operations, where the single-point crossover refers to the exchange of a different gene between two chromosomes, and multipoint crossover refers to the exchange of multiple different genes between two chromosomes. The second step is the intrachromosomal crossover operation, where a larger solution space for movement cost minimization can be obtained by adjusting the task execution route of the participant. The intrachromosomal crossover is achieved by flipping some genes in the chromosome. As shown in Fig.5, Fig.5(a), and Fig. 5(b) represent the interchromosomal crossover. Fig. 5(a) shows that the T_2 gene in chromosome 1 and the T_4 gene in chromosome 2 have achieved a single-point crossover. Fig.5(b) shows that the T_2 and T_5 genes in chromosome 1 have achieved multi-point crossover with the T_4 and T_6 genes in chromosome 2, respectively. Fig.5(c) represents the intrachromosomal crossover. Fig. 5(c) shows that the sequence of T_4, T_1, T_3 genes in chromosome 1 has changed to T_3, T_1, T_4 after the intrachromosomal crossover.

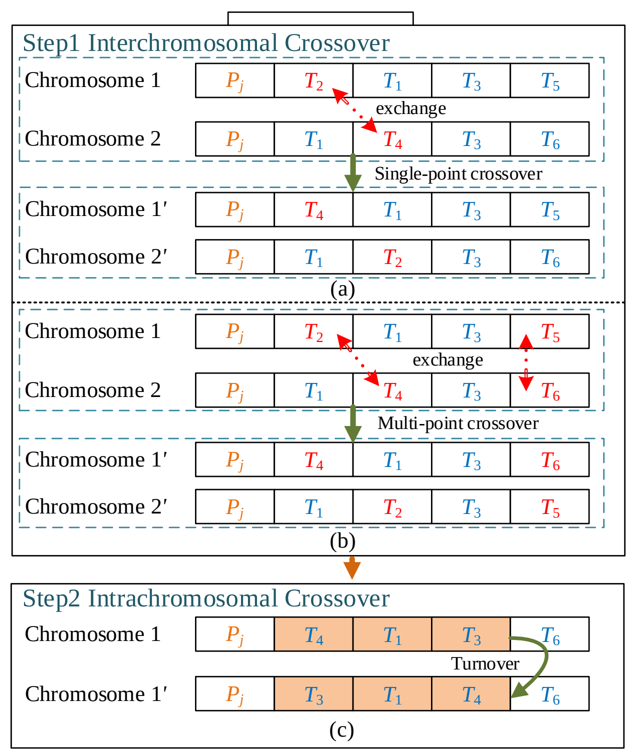

Fig.5. Crossover operation

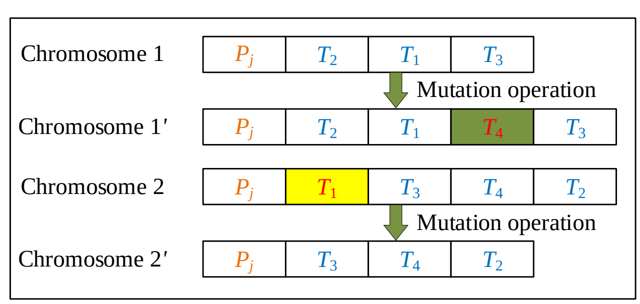

Fig.6. Mutation operation.

Mutation operation. The mutation operation generates a new chromosome by changing the genes in the chromosome, which can increase the diversity of the population and avoid falling into a local optimum effectively. In this paper, the mutation operation is achieved by adding or removing task genes from the chromosome. For example, as shown in Fig.6, a new gene T_4 is added to chromosome 1, and a gene T_1 is removed from chromosome 2. It is worth noting that since a task is allocated to the participant at most once, it is necessary to avoid adding a task that already exists in the chromosome. In addition, for a chromosome that contains all tasks in the candidate set, adding a task is not allowed. Similarly, for a chromosome that contains only one task, removing the task is also not allowed.

Based on the above concepts, the steps of the online task allocation algorithm can be described as follows. i) Generate an initial population with N chromosomes, where each chromosome represents a task allocation scheme. ii) According to the designed fitness function, calculate the fitness of all chromosomes in the population. iii) Generate a new generation population through selection, crossover, and mutation operations. iv) Repeat the steps ii) and iii) until the maximum number $\mathcal{T}$ of iterations is reached, and output the chromosome with the largest fitness as the optimal solution. The demonstration of these steps is described in detail in Algorithm 1.

Algorithm 1 Online Task Allocation Algorithm Based on An Improved Genetic Algorithm

Input: The participant P_j and its candidate set T_C^j.
Output: A best chromosome, i.e., a best task allocation scheme for participant P_j.

1: Generate an initial population POP_0 with N chromosomes.
2: Set iteration counter $t = 0$.
3: **Repeat**
4: $POP'_t \leftarrow$ select chromosomes by using the selection operations on POP_t.
5: $POP''_t \leftarrow$ generate new chromosomes by using interchromosomal crossover and intrachromosomal crossover operations on POP'_t.
6: $POP'''_t \leftarrow$ generate new chromosomes by using mutation operations on POP''_t.
7: $POP_{t+1} \leftarrow POP'''_t$.
8: $t \leftarrow t + 1$.
9: **Until** t = the maximum number of iterations $\mathcal{T}$.

In Algorithm 1, given the participant P_j and its candidate set T_C^j with c tasks, the computation complexity of population initialization is $O(Nc)$. Since the algorithm iterates $\mathcal{T}$ times, the computation complexity of selection operation is $O(N\mathcal{T})$, the computation complexity of crossover operation is $O((N/2+N)\mathcal{T})$, and the computation complexity of mutation operation is $O(N\mathcal{T})$. Since N and $\mathcal{T}$ are constants, the maximum computation complexity of the designed algorithm is $O(Nc) + O(N\mathcal{T}) + O((N/2+N)\mathcal{T}) + O(N\mathcal{T}) = O(c)$.

V. Performance Evaluation

In this section, simulations are conducted by using real data sets to predict the quantity of available participants in FTSA. In the below simulations, we use the root mean square error and the mean absolute error to analyze the accuracy of the prediction. In the experiments of online task allocation, the platform utility and the movement cost of participant are collected and compared by rounds of simulations.

The simulations mainly have the following two purposes: (1) to verify the effect of the FTSA method in terms of the accuracy of prediction and (2) to prove the effect of OTAGA in terms of maximizing the platform utility and minimizing the movement cost of participant.

A. Experimental Setting

1) Data set and Model Settings: In this paper, the data set Geolife[33] is used, which was gathered in the Geolife project (Microsoft Research Asia) by 182 participants in a period of over three years (from April 2007 to August 2012). This data set contains 17,621 trajectories with a total distance of about 1.2 million kilometers and a total duration of 48,203+ hours. The GPS trajectories in this dataset are represented by time-stamped points, each of which contains the latitude and longitude. In our simulations, we select participants in the area with the northern latitude from 39.975 to 40.025 and eastern longitude from 116.31 to 116.35, and evenly divide the area into 10 subareas. In order to facilitate the calculation of the movement cost of the participant, we represent the location of the participant and the tasks in the form of coordinates.

The simulation parameters are summarized in Table II.

Table II
Simulation Parameters

Symbol	Description	Value
$\boldsymbol{h_{i,j}}$	The willingness of participant P_j to execute task T_i	0~1
$\boldsymbol{\varepsilon_{i,j}}$	The sensing ability of participant P_j to execute task T_i	0~1
$\boldsymbol{g_{i,j}}$	The expected reward of participant P_j to execute task T_i	5~15
$\boldsymbol{w_{i,j}}$	The sensor requirement of task T_i for participant P_j	{0,1}
$\boldsymbol{Time_j}$	The remaining time of participant P_j to execute task	0~120 min
$\boldsymbol{uc_j}$	The movement cost per unit distance of participant P_j	1
$\boldsymbol{dc_j}$	The unit consumption cost of participant p_j 's intelligent device	1

2) Benchmark Algorithms: In order to verify the accuracy of FTSA in predicting the quantity of available participants, we compare it with TSP-SVM, which is a time series prediction method based on SVM.

- *Time Series Prediction Method Based on SVM (TSP-SVM)*: This algorithm uses the support vector machine method in machine learning to predict future data through historical data.

Moreover, in order to emphasize the advantages of the designed model and algorithm in terms of maximizing the platform utility and minimizing the movement cost of participant, we utilize the following baseline task allocation algorithms for comparative studies:

- *Random Allocation Algorithm (RA):* This algorithm randomly allocates tasks to the participants who are in a specific time and space.
- *Utility-Greedy Allocation Algorithm (UGA):* In this algorithm, the platform will allocate tasks which can lead the highest utility for the participants in the special spatiotemporal area until the remaining activity time of the participants is exhausted.
- *Cost-Greedy Allocation Algorithm (CGA):* In this algorithm, the platform will allocate task which can lead the lowest movement cost for the participant. When the remaining activity time of the participants is exhausted, the whole process of the task allocation is over.

B. *Performance Evaluation and Discussion*

The comparison results between the proposed method and the aforementioned benchmark algorithms on the accuracy of prediction, the platform utility, and the movement cost of the participant are shown in this section. In order to avoid the influence of randomness on the effectiveness of experimental results, simulation data is obtained by computing the average values from 100 runs for each test under the same simulation environment.

1) The accuracy of prediction.

The accuracy of the prediction is an important criterion for evaluating the prediction algorithm. In this paper, we use the root mean square error and mean absolute error between the value of the predicted and the true quantity to evaluate the accuracy of the prediction.

In our simulations, the quantity of available participants for a task on a day is predicted by analyzing historical data of the task on 12-days before the day.

The root mean square error is one metric in our simulations to measure the prediction accuracy, because it reflects the deviation from the predicted value to the true value in algorithms. When the root mean square error is smaller, the deviation between the predicted value and the true value is smaller. In Table III, the root mean square errors between the predicted value and the true value when the total number of tasks is 10, 20, 30, 40, and 50 are shown, which tells us that, the prediction accuracy of FTSA is better than that of TSP-SVM.

Table III
The root mean square error between the predicted value and the true value when use different methods for different number of tasks

	10	20	30	40	50
FTSA	2.14	3.32	3.45	3.57	3.59
TSP-SVM	8.9	7.79	6.83	7.55	7.34

The mean absolute error is the other important indicator to judge the prediction accuracy in our simulations, because it represents the average value of the absolute error between the predicted value and the observed value. Like the root mean square error, the smaller the absolute average error, the higher the accuracy of the prediction algorithm. Table VI shows the mean absolute errors between the predicted value and the true value when the total number of tasks is 10, 20, 30, 40, and 50, and also verifies that the prediction accuracy of FTSA is better than that of TSP-SVM.

Table VI
The mean absolute error between the predicted value and the true value when use different methods for different number of tasks

	10	20	30	40	50
FTSA	1.6	3	2.57	2.65	2.7
TSP-SVM	6.6	6.2	4.77	5.77	5.48

2) The platform utility.

In this part, we compare the platform utility of OTAGA proposed in this paper with that of the other three benchmark algorithms and analyze the factors influencing the platform utility in OTAGA, including the number of tasks, number of iterations and the remaining activity time of the participants.

Fig.7 shows the influence of the number of tasks on the platform utility in OTAGA and the other three benchmark algorithms. In more detail, in all of the four algorithms, when the number of tasks increases, the platform utility tends to increase. This is because, when the number of tasks increases, there will be more tasks that can make the platform utility higher. Therefore, these algorithms will allocate tasks with a higher utility to the participant to maximize the platform utility.

In addition, Fig.7 tells us that UGA can achieve higher platform utility than CGA and RA. The reason is that UGA always selects tasks with higher utility. However, UGA does not take other attributes of the task (the time required to execute the task, etc.) into consideration, so it will fall into a local optimum. In OTAGA, there are operations such as crossover and mutation to avoid the obtained task allocation set from falling into the local optimal situation. Therefore, OTAGA is superior to the other three benchmark algorithms in maximizing the platform utility.

Fig.8 shows the influence of the number of iterations on the platform utility in OTAGA with different numbers of tasks. From Fig.8, we can observe that, regardless of the number of tasks, the platform utility will increase as the number of iterations increases, and finally it will converge to an optimal value. This is because, as the number of iterations increases, a better task allocation scheme can be found, and after a certain number of iterations, the task allocation scheme with the maximum platform utility is obtained.

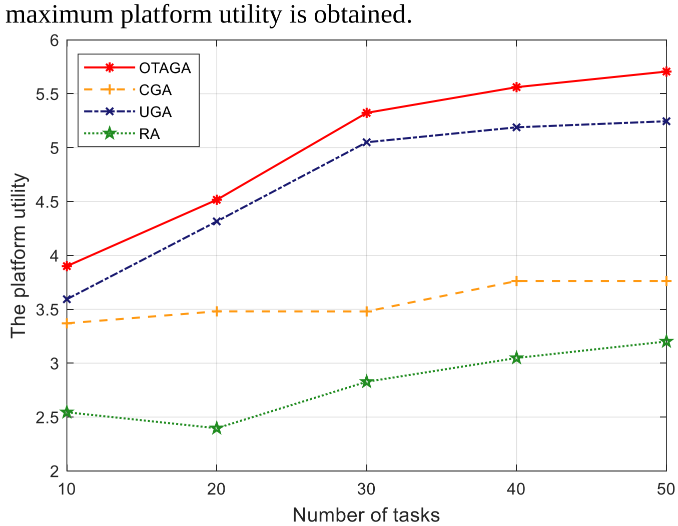

Fig.7 Comparison of the platform utility between OTAGA and other benchmark algorithms when the number of tasks changes (Number of iterations = 100, The remaining activity time = 120min).

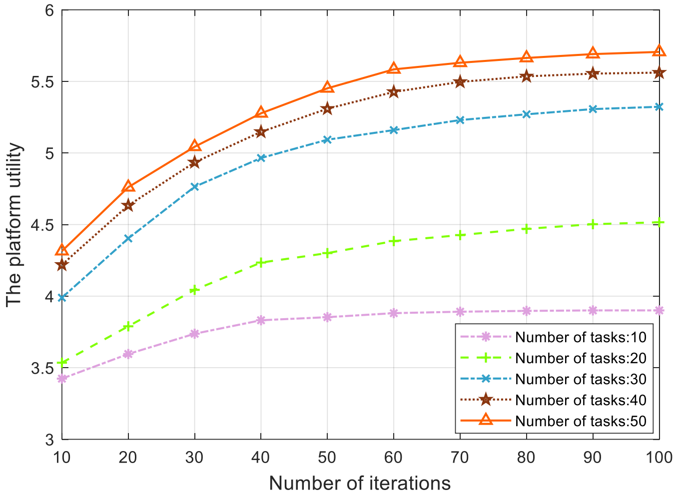

Fig.8 Utility of platform when the number of tasks changes (Algorithm: OTAGA, The remaining activity time = 120 min).

Fig.9 shows the influence of the participants' remaining activity time on the platform utility when the number of tasks is fixed at 20 in OTAGA and the other three benchmark algorithms. In other words, regardless of the allocation algorithm, the platform utility will increase as the remaining activity time of the participants increases. And regardless of the remaining activity time of the participants, OTAGA can get a higher platform utility. This is because when a participant's remaining activity time to perform tasks increases, it can perform more tasks, which is beneficial to increase the platform utility.

Fig.10 shows the influence of the number of iterations on the platform utility when the remaining activity time is different and the number of tasks is fixed at 20 in OTAGA. It can be analyzed from Fig.10 that, no matter how much the remaining activity time the participants, as the number of iterations increases, the platform utility also increases and eventually converges. It is because, as the number of iterations increases; a better task allocation scheme can be found and the platform utility increases.

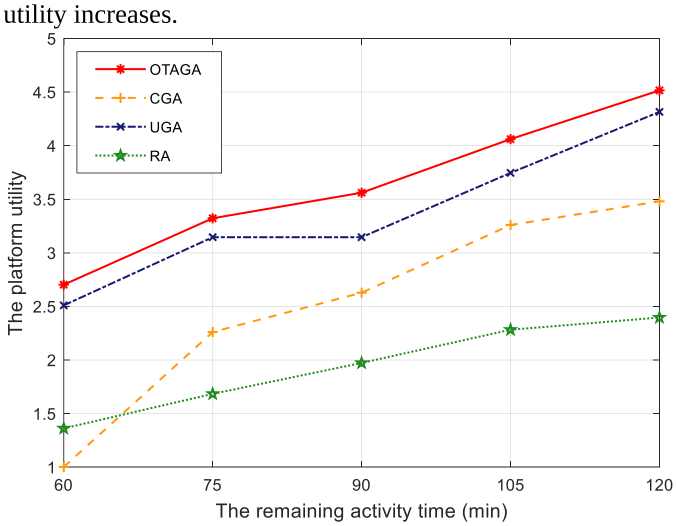

Fig.9 Comparison of utility of platform between OTAGA and other benchmark algorithms when the remaining time of participant changes (Number of tasks = 20).

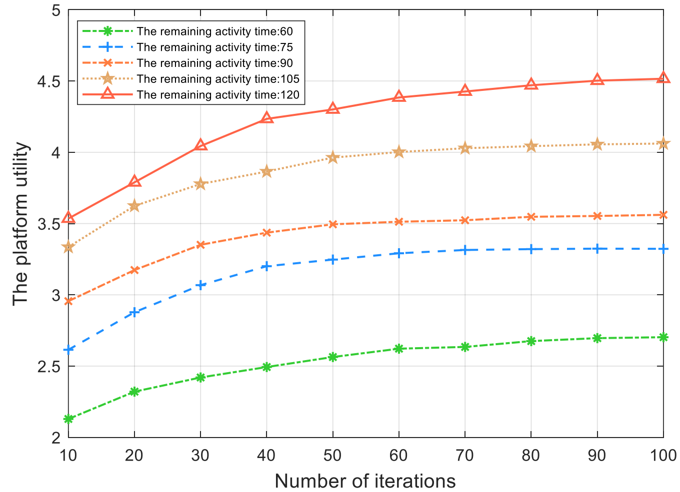

Fig.10 Utility of platform when the number of iterations changes (Algorithm: OTAGA, Number of tasks = 20).

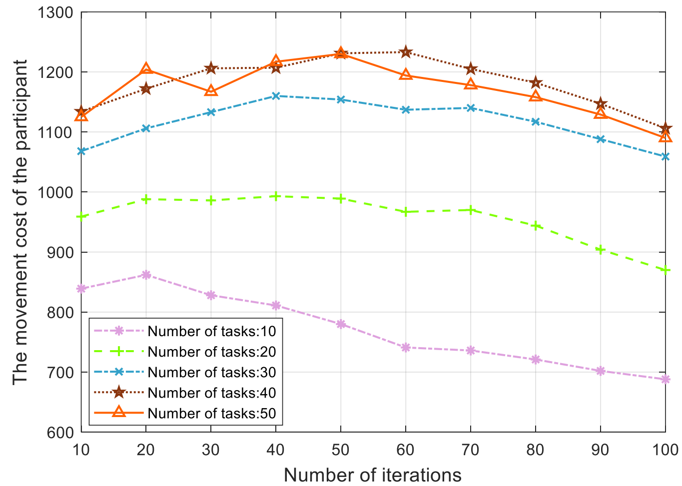

Fig.11 The movement cost of the participant when the number of iterations changes (Algorithm: OTAGA).

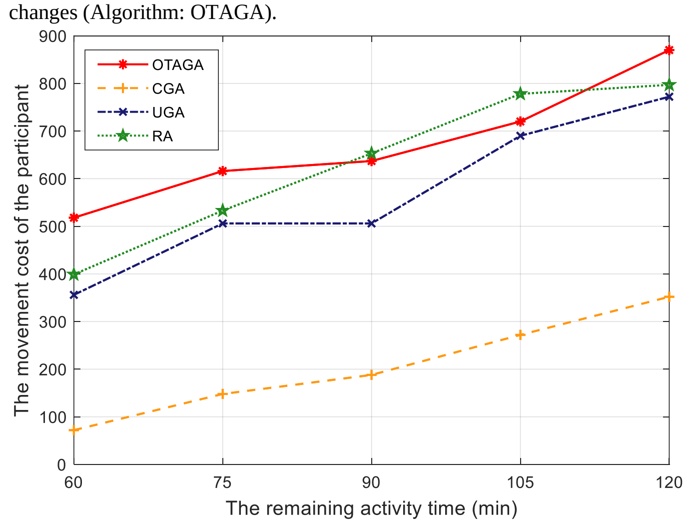

Fig.12 Comparison of the movement cost of the participant between OTAGA and other benchmark algorithms when the remaining activity time of participant changes (Number of tasks = 20).

3) *The movement cost of the participant.*

In this part, we compare the movement cost of the participant of OTAGA proposed in this paper with that of the other three benchmark algorithms and analyze the factors influencing the movement cost of the participant in OTAGA, including the number of iterations and the remaining activity time of the participants.

Fig.11 shows the influence of the number of iterations on the movement cost of the participant when the number of tasks is different. As shown in Fig.11, it can be observed that as the number of iterations increases, the movement cost of the participant will first increase, and then when the iterations continue, the movement cost of the participant begins to gradually decrease and tends to be stable. The reason for the increase in the movement cost of the participant in the initial iteration is that the task allocation set obtained at the beginning cannot maximize the platform utility. Therefore, after the interchromosomal crossover and mutation operations, the tasks and the number of tasks in the task allocation set will change, which causes the increase in the movement cost of the participant. The reason for the subsequent gradual decline is that when the optimal task allocation set is determined, the platform utility stabilizes after a certain number of iterations. The main operation of the algorithm is the intrachromosomal crossover operation. This operation changes the order of execution of the tasks, so that while maximizing the utility, the movement cost of the participant is minimized.

Fig.12 shows the influence of the remaining activity time on the movement cost of the participant in different algorithms when the number of tasks is fixed at 20. In Fig.12, it can be observed that as the remaining activity time of participant increases, the movement cost of the participant also increases. The reason for this phenomenon is that the more remaining activity time, the more tasks participants can perform, so the movement cost will increase accordingly. In addition, it can be obviously seen in the figure that the participant has the smallest movement cost in the CGA, but at the same time, the platform utility obtained by this algorithm is also the smallest. On the contrary, the OTAGA can maximize the platform utility while minimizing the movement cost of the participant, which is more in line with actual needs.

4) OTAGA performance analysis

Generally, genetic algorithm stops iteration by setting the maximum number of iterations or presetting the objective function value. In this paper, we set the maximum number of iterations as the condition for stopping the algorithm. Through a lot of experiments, we find that when the maximum number of iterations is set to 100, the OTAGA can make the objective function reach the approximate optimal solution. The improvement of OTAGA is mainly in chromosome representation and operator, so that it can solve the online task allocation problem proposed in this paper, but its performance in terms of time consumption is not greatly improved. Fig.13 shows the average time consumption of OTAGA under different number of iterations when the number of tasks is set to 50.

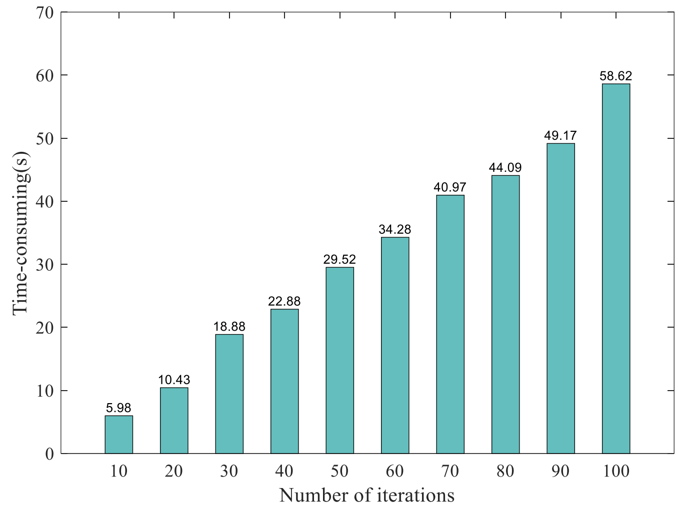

Fig.13 Time consuming when the number of iterations changes (Algorithm: OTAGA, Number of tasks = 50)

In summary, in terms of the accuracy of prediction the quantity of available participants, the FTSA method proposed in this paper has higher prediction accuracy than TSP-SVM. In terms of task allocation, although the OTAGA's performance in terms of time consumption is not greatly improved, it has an excellent performance in maximizing the platform utility when compared with other benchmark algorithms. And while maximizing the platform utility, it can also minimize the movement cost of the participant.

VI. Conclusion

The existing studies of online task allocation lack deep consideration regarding the impact of the participant quantity on the quality and efficiency of task completion. In this paper, we investigate a participant-quantity-aware online task allocation problem. In view of the difficulty of predetermining the participant quantity in MCS, a FTSA method is developed to predict the participant quantity available for each task in a specific time and space. According to the predicted participant quantity, two reasonable attributes for each task, including the task's threshold on participants' sensing ability and the reward provided for participants to execute the task, can be calculated separately. On this basis, we design an online task allocation algorithm based on an improved genetic algorithm to allocate an appropriate set of tasks to each participant who arrives in real-time, so as to maximize the platform utility and minimize the movement cost of the participant. Finally, the effectiveness of the FTSA method is validated by comparing it with time series prediction method based on SVM. The effectiveness of the online task allocation algorithm based on an improved genetic algorithm is validated by comparing with three baseline task allocation algorithms. Simulation results demonstrate that the prediction algorithm proposed in this paper has a high accuracy rate. The task allocation algorithm also performs well in increasing the platform utility and reducing the movement cost of the participant.

References

[1] R. K. Ganti, F. Ye and H. Lei, "Mobile crowdsensing: current state and future challenges," in IEEE Communications Magazine, vol. 49, no. 11, pp. 32-39, November 2011, DOI: 10.1109/MCOM.2011.6069707.

[2] M. R. Ra, B. Liu, T. F. La Porta, et al, "Medusa: A programming framework for crowd-sensing applications," in *Proc.* 10th international conference on Mobile systems, applications, and services, New York, NY, USA, 2012, pp. 337–350.

[3] P. Simoens, Y. Xiao and P. Pillai, "Scalable crowd-sourcing of video from mobile devices," in *Proc.* 11th annual international conference on Mobile systems, applications, and services. New York: Association for Computing Machinery, 2013, pp. 139–152.

[4] T. Das, P. Mohan and V. N. Padmanabhan, "PRISM: Platform for remote sensing using smartphones," in *Proc.* 8th international conference on Mobile systems, applications, and services. New York: Association for Computing Machinery, 2010, pp. 63–76.

[5] WAZE. [Online]. Available: https://www.waze.com/. Accessed on: Apr. 19, 2021.

[6] Y. Liu, Z. Yu, B. Guo, Q. Han, J. Su and J. Liao, "CrowdOS: A Ubiquitous Operating System for Crowdsourcing and Mobile Crowd Sensing," in IEEE Transactions on Mobile Computing, vol. 21, no. 3, pp. 878-894, 1 March 2022, DOI: 10.1109/TMC.2020.3015750.

[7] G. Sun, L. Song and H. Yu, "A Two-tier Collection and Processing Scheme for Fog-based Mobile Crowd Sensing in the Internet of Vehicles," in IEEE Internet of Things Journal, vol. 8, no. 3, pp. 1971-1984, 2021.

[8] M. Kim and Kim Y. Kim, "Multi-Blockchain Structure for a Crowdsensing-Based Smart Parking System," in Future Internet, vol.12, no.5, pp. 90, 2020.

[9] D. Wu, T. Xiao and X. Liao, "When Sharing Economy Meets IoT: Towards Fine-grained Urban Air Quality Monitoring through Mobile Crowdsensing on Bike-share System, " in ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 4, no. 2, pp. 1-26, 2020.

[10] J. Wang, L. Wang and Y. Wang, "Task allocation in mobile crowd sensing: State-of-the-art and future opportunities", in IEEE Internet of Things journal, vol. 5, no.5, pp. 3747–3757, 2018.

[11] W. Zhu, W. Guo, and Z. Yu. "Social-Aware Task Allocation in Mobile Crowd Sensing," in Wireless Communications and Mobile Computing pp. 1-15, 2020.

[12] C. Lai and X. Zhang, "Duration-Sensitive Task Allocation for Mobile Crowd Sensing," in IEEE Systems Journal, vol. 14, no. 3, pp. 4430–4441, Sept. 2020, DOI: 10.1109/JSYST.2020.2967847.

[13] X. Li and X. Zhang, "Multi-Task Allocation Under Time Constraints in Mobile Crowdsensing," in IEEE Transactions on Mobile Computing, vol. 20, no. 4, pp. 1494–1510, 1 April 2021, DOI: 10.1109/TMC.2019.2962457.

[14] C. Dai, X. Wang, K. Liu, D. Qi, W. Lin and P. Zhou, "Stable Task Assignment for Mobile Crowdsensing With Budget Constraint," in IEEE Transactions on Mobile Computing, vol. 20, no. 12, pp. 3439-3452, 1 Dec. 2021, DOI: 10.1109/TMC.2020.3000234..

[15] X. Chen, "A Stable Task Assignment Scheme in Crowdsourcing," in *Proc.* 2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), New York, NY, USA, 2019, pp. 489–494, DOI: 10.1109/CSE/EUC.2019.00096.

[16] X. Zhang, J. Ding, X. Li, T. Yang, J. Wang and M. Pan, "Mobile Crowdsensing Task Allocation optimization with Differentially Private Location Privacy," in *Proc.* ICC 2020 - 2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 2020, pp. 1–6, DOI: 10.1109/ICC40277.2020.9149398.

[17] M. Xiao, M. Li, L. Guo, M. Pan, Z. Han and P. Li, "Securing Task Allocation in Mobile Crowd Sensing: An Incentive Design Approach," in *Proc.* 2019 IEEE Conference on Communications and Network Security (CNS), Washington, DC, USA, 2019, pp. 19–27, DOI: 10.1109/CNS.2019.8802697.

[18] N. Song, D. Lu, Y. Shi, G. Zhang and H. Liu, "Collaborative Teams Recruitment Based on Dual Constraints of Willingness and Trust for Crowd Sensing," in *Proc.* 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD), 2022, pp. 1251-1256, doi: 10.1109/CSCWD54268.2022.9776290.

[19] C. H. Liu, B. Zhang, X. Su, J. Ma, W. Wang and K. K. Leung, "Energy-Aware Participant Selection for Smartphone-Enabled Mobile Crowd Sensing," in IEEE Systems Journal, vol. 11, no. 3, pp. 1435-1446, Sept. 2017, DOI: 10.1109/JSYST.2015.2430362.

[20] W. Jiaying, Z. Xiaoyu, M. Xingxing, C. Zhen and K. Wenshan, "User Willingness-Based participant selection strategy of Crowdsensing," in *Proc.* 2022 Asia Conference on Algorithms, Computing and Machine Learning (CACML), 2022, pp. 809-816, DOI: 10.1109/CACML55074.2022.00139.

[21] Y. Kang, X. Miao, K. Liu, L. Chen and Y Liu, "Quality-Aware Online Task Assignment in Mobile Crowdsourcing," in *Proc.* 12th International Conference on Mobile Ad Hoc and Sensor Systems, Dallas, TX, USA, 2015, pp. 127–135, DOI: 10.1109/MASS.2015.40.

[22] H. Li, T. Li, W. Wang and Y. Wang, "Dynamic Participant Selection for Large-Scale Mobile Crowd Sensing," in IEEE Transactions on Mobile Computing, vol. 18, no. 12, pp. 2842–2855, 1 Dec. 2019, DOI: 10.1109/TMC.2018.2884945.

[23] D. Li, J. Zhu and Y. Cui, "Prediction-Based Task Allocation in Mobile Crowdsensing," in *Proc.* 15th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN), Shenzhen, China, 2019, pp. 89–94, DOI: 10.1109/MSN48538.2019.00029.

[24] J. Wang, "HyTasker: Hybrid Task Allocation in Mobile Crowd Sensing," in IEEE Transactions on Mobile Computing, vol. 19, no. 3, pp. 598-611, 1 March 2020, DOI: 10.1109/TMC.2019.2898950.

[25] Y. Li, B. Zhu, T. Huang and D. Zhu, "Mota: Multi-Stage Multi-Task Online Assignment Algorithm Based on Opportunistic Crowdsensing," in *Proc.* 16th International Computer Conference on Wavelet Active Media Technology and Information Processing, 2019, pp. 345-348, DOI: 10.1109/ICCWAMTIP47768.2019.9067694.

[26] S. Bradai, S. Khemakhem and M. Jmaiel, "Re-OPSEC: Real time opportunistic scheduler framework for energy aware mobile crowdsensing," in *Proc.* 24th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), 2016, pp. 1-5, DOI: 10.1109/SOFTCOM.2016.7772174.

[27] X. Wang, R. Jia, X. Tian, X. Gan, L. Fu and X. Wang, "Location-Aware Crowdsensing: Dynamic Task Assignment and Truth Inference," in IEEE Transactions on Mobile Computing, vol. 19, no. 2, pp. 362-375, 1 Feb. 2020, DOI: 10.1109/TMC.2018.2878821.

[28] G. Sun, Y. Wang, X. Ding and R. Hu, "Cost-Fair Task Allocation in Mobile Crowd Sensing With Probabilistic Users," in IEEE Transactions on Mobile Computing, vol. 20, no. 2, pp. 403–415, 1 Feb. 2021, DOI: 10.1109/TMC.2019.2950921.

[29] C. Bassem, "Forgive But Don't Forget: On Reliable Multi-Task Allocation in Mobile CrowdSensing Platforms," in *Proc.* 2020 IEEE International Conference on Smart Computing (SMARTCOMP), Bologna, Italy, 2020, pp. 98–105, DOI: 10.1109/SMARTCOMP50058.2020.00033.

[30] He, X., M. Liu, and G. Yang. "Spatiotemporal opportunistic transmission for mobile crowd sensing networks," in Personal and Ubiquitous Computing (2020):1-11.

[31] S. D. Immanuel and U. K. Chakraborty, "Genetic Algorithm: An Approach on Optimization," in *Proc.* 2019 International Conference on Communication and Electronics Systems (ICCES), 2019, pp. 701-708, DOI: 10.1109/ICCES45898.2019.9002372.

[32] X. Yan, and J. Han. "gSpan: Graph-Based Substructure Pattern Mining." in *Proc.* 2002 IEEE International Conference on Data Mining, Maebashi City, Japan, 2002, pp. 721-724, DOI: 10.1109/ICDM.2002.1184038.

[33] Geolife GPS Trajectories. Accessed: Jul. 1, 2020. [Online]. Available: http://research.microsoft.com/en-us/downloads/b16d359d-d164-469e-9fd4-daa38f2b2e13/

Guisong Yang received his Ph.D. degree in Control Theory and Control Engineering from TongJi University in 2013, and worked as a research scholar at Michigan State University from 2009 to 2011. Currently, He is an associate professor with the Department of Computer Science and Engineering at University of Shanghai for Science and Technology. His research interests include Internet of Things and pervasive computing, delay tolerant and opportunistic networks, and mobile crowd sensing. He is a member of IEEE and ACM.

Dongsheng Guo received the B.S. degree in software engineering from Nanjing Forestry University, Nanjing, China, in 2020. He received the master’s degree in computer science and technology at the University of Shanghai for Science and Technology, Shanghai, China, in 2023. His research interest is mobile crowd sensing.

Buye Wang received the B.S. degree in information management and information system from Tianjin University of Technology, Tianjin, China, in 2017. He received the master’s degree in computer science and technology at the University of Shanghai for Science and Technology, Shanghai, China, in 2021. His research interest is mobile crowd sensing.

Xingyu He received her Ph.D. degree in Control Theory and Control Engineering from TongJi University in 2017. She is an associate professor with College of Communication and Art Design at University of Shanghai for Science and Technology. Her research interests include wireless sensor networks and pervasive computing, delay tolerant networks, swarm intelligence and mobile crowd sensing.

Jiangtao Wang received the PhD degree from Peking University, China, in 2015. He is currently an Associate Professor with Tenure in the Intelligent Healthcare Center, Coventry University, UK. Before that, he was a lecturer with the School of Computing and Communications at University, UK. His research interest includes mobile and pervasive computing, crowdsensing/crowdsourcing, and IoT.

Gang Wang received the B.S., M.S., and Ph.D., all in electrical engineering, in 2005, 2008, and 2014, respectively, from the Northwestern Polytechnical University, Xi'an, China, Beihang University (with highest honor), Beijing, China, and University of Arkansas at Fayetteville, USA, respectively. He is a senior staff communications system engineer at MaxLinear Inc, California, USA, and previously served as a senior DSP engineer at PCTEL, Inc., USA, a lead scientist at IFT, Inc., USA, and wireless system engineer in New Postcom Equipment Co., Ltd, in Beijing. His current research interests are machine learning and deep learning applications for 5G and 6G, IoT applications for smart city, and robust cognitive radio space communications. He holds 8 patents worldwide and published over 30 papers including JSAC, TWC, TVT, ICC and Globecom.