

Service Allocation in Practical Sensor Networks by Enhanced Fuzzy Clustering Method

¹G Naga Pavani, ²SK Pujitha, ³Bejjanki Pradeep Kumar, ⁴P Pavan Kumar
^{1,2} Assistant Professor, Dept of CSE, Vignan's Lara Institute of Technology &
Science, A.P., India

³Assistant Professor, Dept of IT, Vignan's Lara Institute of Technology & Science,
A.P., India

⁴Assistant Professor, Dept of CSE, Vignan's Lara Institute of Technology &
Science, A.P., India

Email: yaramsettipavani@gmail.com, pujithapu533@gmail.com,
bpkn@rocketmail.com, pavanathraya@gmail.com

Abstract

Mobile crowd sensing (MCS) structures normally entice numerous members with widely varying sensing fees and hobby opportunities to perform responsibilities, wherein correct challenge task performs an essential characteristic and additionally faces many demanding situations (e.g., a way to simplify the complicated challenge task method and decorate matching accuracy among obligations and members, even as ensuring submitted facts credibility). To conquer those challenges, we propose a provider gain conscious multi-project challenge (SBAMA) technique in this paper. Firstly, service benefits of individuals are modeled primarily based totally on their task trouble, mission history, sensing capability, and sensing positivity to satisfy differentiated necessities of several undertaking sorts. Subsequently, customers are then clustered through better fuzzy clustering approach. Finally, a gradient descent set of rules is designed to inform venture types to participants accomplishing the maximum carrier benefit. Simulation effects confirm that the proposed challenge project approach no longer best correctly reduces matching complexity but additionally improves challenge finishing touch fee.

Keywords: Mobile Crowd Sensing; task assignment; service benefit; users clustering; gradient descent algorithm

1. Introduction

The diversification and popularization of embedded mobile devices enable innumerable user-centric mobile crowd sensing (MCS) applications (e.g., traffic monitoring, pollution monitoring, and indoor positioning) [1–9]. A typical MCS system includes two entities (i.e., users and platform), where users not only publish tasks to acquire information from the platform but also collect sensing data for the platform. As a bridge between task publishers and participants, the platform helpfully selects suitable participants to complete tasks for publishers. Generally, sensing tasks in a MCS system are allocated to multiple participants and accomplished cooperatively. It is crucial for a MCS system to provide task publishers with reliable services. The task sensing process of MCS system relies on massive participants whose sensing positivity and sensing capacity are diverse for different tasks. If a task is randomly assigned to participants, the quality of its results may be severely affected, the credibility of collected sensing data may be reduced, and the corresponding computing resource consumption may increase. Therefore, task assignment strategies should be reasonably designed to eliminate the above uncertainties, which can

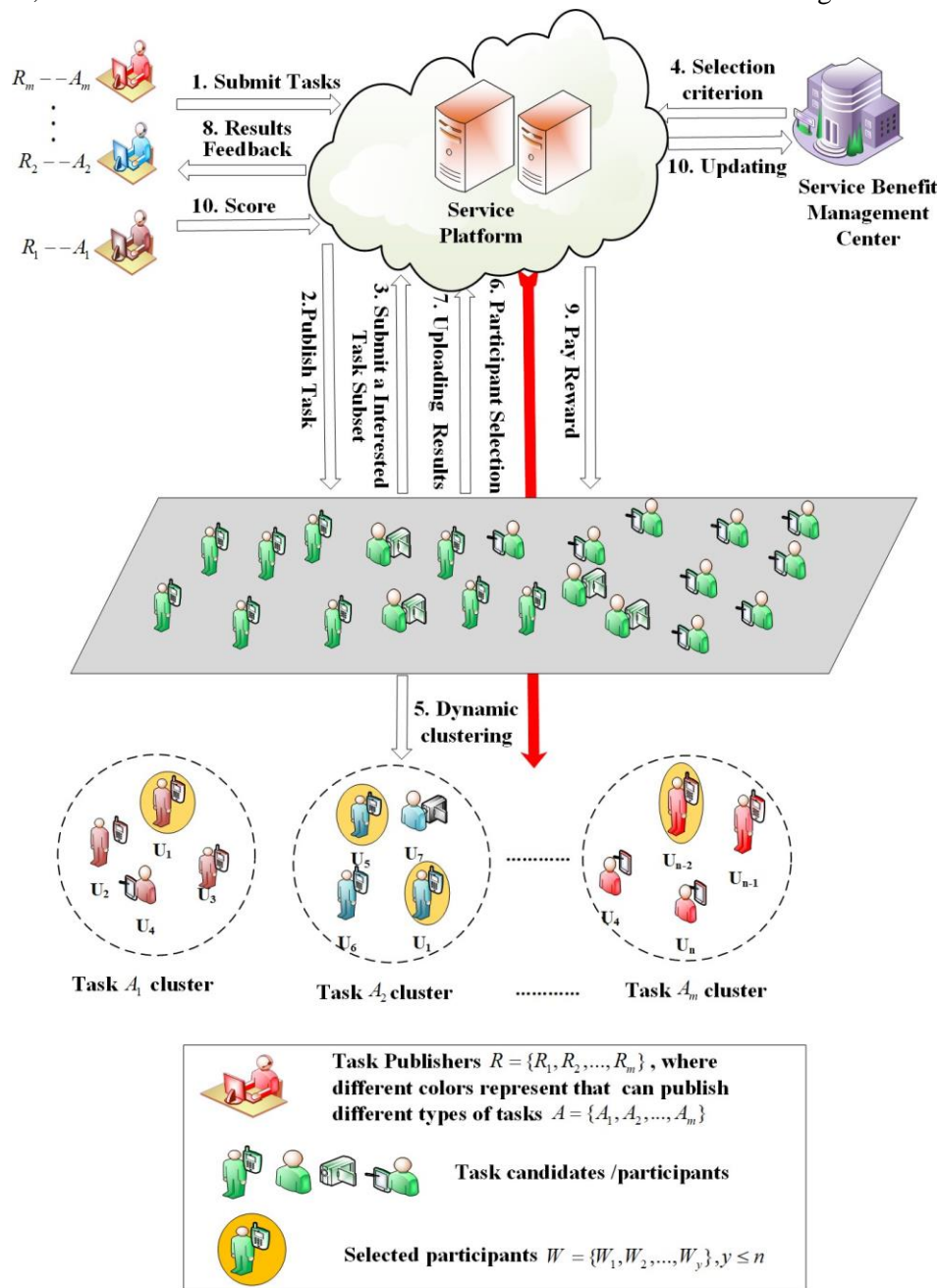
dramatically enhance the philosophy behind MCS participant collaborations. Technically, how to objectively evaluate the matching accuracy of task assignment strategies is a major challenge and remains in dispute. In response to this challenge, many studies presented solutions using varying methods and with different emphases. Jin, H et al. assigned MCS tasks to participants according to their capabilities to maximize sensing coverage. The authors of assigned tasks based on the history of submitted high-quality results. However, they had different definitions of data quality. Yue, W et al. mainly considered the coverage quality of sensing results, while Sabrina, K.N.M et al. measured the data quality based on the actual evaluation of data. Liu, S et al. evaluated participant's service quality based on its context and cost, and developed a Modified Thompson Sampling Worker Selection (MTS-WS) algorithm to select workers in a reinforcement learning manner. Addressing the problem of performance maximization in MCS, a context-aware hierarchical online learning algorithm was proposed in. In detail, a local controller (LC) in the mobile device of a participant regularly observed the participant's context, based on which the participant's context-specific performance could be estimated and the participants could be selected. Although the authors of considered multiple task-related factors, there are still some limitations and factors that may intuitively affect MCS performance (e.g., task difficulty, task history, sensing capacity, and sensing positivity were omitted). Besides, the quality of data uploaded for different task types was not clearly modeled in these studies. Consequently, it may be impossible for the platform to assign tasks reasonably and accurately, resulting in low matching accuracy/task completion rate, high computing resource consumption, and incredibility of data. Focusing on above limitations, we propose a service benefit aware (SBAMA) multi-task assignment strategy for MCS. The service benefit of participants is first modeled. Subsequently, we propose an enhanced Fuzzy C-Means (FCM) algorithm to dynamically cluster users in terms of their task preferences. Finally, an iterative participant search method based on gradient descent is designed to match participants with the best service benefit in each cluster quickly and accurately. The proposed SBAMA multi-task assignment strategy offers an interest tradeoff between participants and platform given a fixed budget and a certain movement distance. Simulation results verify that SBAMA can quickly and accurately find the most appropriate participants for all types of tasks. Therefore, task completion rate based on the proposed strategy is convincingly high. The main contributions of our paper can be summarized as follows: (1) A service benefit evaluation model is established, from several different perspectives of tasks and participants, to comprehensively interpret impacts of task difficulty, task history, sensing capacity, and sensing positivity on service benefit received by the platform. (2) An enhanced FCM algorithm is designed to cluster users. Specifically, a task preference threshold allows participants to join more than one cluster if such participants have similar task preferences. The generated clusters can effectively reduce the time consumption of the optimization problem while increasing participant matching accuracy. (3) An iterative gradient descent algorithm is proposed to tune the tradeoff between interests of participants and platform. Particularly, it decouples the service benefit from movement distance such that the most appropriate participants for tasks can be found accurately and quickly. The rest of this paper is organized as follows. Related works are introduced in Section 2. Section 3 presents the task assignment framework. The service benefit model is proposed in Section 4. Section 5 elaborates on the task assignment strategy. Simulation results validating our proposed SBAMA are given in Section 6.

Finally, Section 7 concludes this paper and discusses future work. 2. Related Works In recent years, task assignment for MCS systems has been attracting increasing research attention. Wei, G et al. proposed a heterogeneous multi-task allocation mechanism based on spatiotemporal correlation. The reference and non-reference tasks were distinguished through utilizing granularity settings based on which the best triple (i.e., worker-cycle-region) was obtained. Besides, to improve the task assignment efficiency, a decomposition and combination framework was designed in for large-scale scenarios. In, a location-based online task assignment method was proposed under constraints of distance and budget, incorporating quality/progress-based, task-density-based, travel-distance-balance-based, and bio-inspired-travel-distance-balance-based algorithms, to search for the optimal participants maximizing the overall task quality. Wang, J et al. studied the deterministic model and random model of trajectories in vehicle-based MCS and proposed an effective vehicle recruitment algorithm to minimize the overall recruitment cost. However, the authors of only considered either the mobility of participants or the spatiotemporal correlation among tasks, where impacts of task requirements and service benefits were ignored. A failure to consider these critical factors together may result in an inaccurate participant matching during the task assignment process, which reduces the task completion rate. The criterion of participant selection in task assignment has been extensively studied. Considering various factors affecting the task participant selection, a task assignment framework was proposed in. Specifically, a unified estimation function was employed to calculate the feasibility of task assignments and the optimal task assignment using a greedy algorithm was obtained. Wang, L et al. and Alsayasneh, M et al. focused on the context information of participants to enhance the task completion rate and MCS quality. In particular, a diverse task composition scheme was studied in terms of participant personalities to dramatically improve user experience. Mavridis, P et al. inferred skills required for tasks from available skill sets and modeled a hierarchical skill tree to match participants with tasks, which was however computationally intensive and therefore inapplicable to scenarios with massive users and tasks. Besides, the credibility of data submitted by participants could not be guaranteed only based on their skills. Although the authors of assigned tasks to relatively appropriate participants to ensure the data credibility, when evaluating the participant selection, key factors such as participant positivity and task difficulty should be carefully considered for the matching accuracy. Therefore, in this paper, we comprehensively evaluate the service benefits of participants as the matching criterion and propose the SBAMA multi-task assignment strategy to enhance MCS performance.

2. Task Assignment Framework

The proposed SBAMA framework is shown in Figure 1. Since tasks arrive randomly, tasks assigned by the platform can be divided into n identical time intervals, denoted as $t = [t_1, t_2 \dots t_n]$. At the beginning of t_n , task publishers $R = \{R_1, R_2 \dots R_m\}$ submit tasks of different types $A = \{A_1, A_2 \dots A_m\}$ to the platform. Each publisher can only submit one type of tasks within each time interval. Note that each task type contains q subtasks, i.e., $A_i = \{a_{i1}, a_{i2} \dots a_{iq}\}$. Specifically, a task has several requirements (i.e., deadline, location, data format, and ID), denoted as $T_{doc} = \{T_d; |A_i; v; ID\}$, where T_d refers to the time range from task start t_s to task end t_e (i.e., $T_d = [t_s, t_e]$) and participants must submit sensing results before this deadline. The specific locations for subtasks of task A_i are denoted by $|A_i = \{l_{i1}, l_{i2} \dots l_{iq}\}$. Due to different content

types for each task, the format of collected data also varies. Without loss of generality, the data format is identified by $v = [v_1, v_2, \dots, v_k]$, where v_k specifies the content type acceptable for each task. The platform then publishes tasks to all potential participants $U = \{u_1, u_2, \dots, u_n\}$ satisfying task requirements T_{doc} . Task candidates then submit a subset of the received tasks $SA = [A_1, A_2, \dots, A_k]$, where $k \leq m$, to the platform indicating their task preferences. Subsequently, the platform employs task assignment strategy to select participants with high service benefits and low costs to perform tasks. In other words, the optimal task participants $W = W_1, W_2, \dots, W_y$ ($y \leq n$) are found. When tasks are completed, the platform compensates the selected participants according to their costs and task difficulties. Finally, upon receiving feedback from the platform, task publishers score the service quality of the participants. Apparently, the scores should be exploited by the platform as an important reference to evaluate and update the service benefit, which further serves as an indicator for the next rounds of task assignment



Generally, MCS tasks are location-dependent and participants have to travel a certain distance to perform the tasks. Therefore, movement distances are an inevitable cost for participants, i.e., cost C_{ij} of participant u_j performing task A_i should be a function of movement distance d_{ij} and sensing cost c_{ij} . Evidently, C_{ij} is proportional to the distances traveled by participants (i.e., $C_{ij}(d_{ij}) \geq C_{ij}(d_{ij}), \forall d_{ij} \geq d_{ij}$) and its growth rate also should increase with the distance (i.e., $d^2C_{ij}(d_{ij})/dd_{ij} \geq 0$). The cost C_{ij} is defined as follows

$$C_{ij} = a_{ij}d_{ij}^2 + b_{ij}d_{ij} + c_{ij} \quad (1)$$

In Equation (1), $a_{ij} > 0$ and $b_{ij} > 0$ are pre-defined system parameters and c_{ij} is constant. Participants expect to get rewards from the platform after completing tasks. Specifically, the reward depends on the task difficulty and movement distance. Given a difficult task, its price per meter should be high, and prices of different types of tasks are denoted by $P = [P_1, P_2 \dots P_m]$. Besides, the reward should not exceed the budget. Note that participants can perform different subtasks in the same time interval to maximize their incomes, only when their locations are not in conflict. Thus, the income of participants can be easily calculated through deducting the cost from the reward; there holds

$$\psi_j(d_{ij}) = \sum_{i=1}^m [(P_i \cdot d_{ij}) - C_{ij}] \quad (2)$$

For the platform, its profit mainly comes from service benefits contributed by participants. Intuitively, participants with greater service benefits generate more profits for the platform and therefore should be prioritized. However, the growth rate of the profit should be slowly attenuated because the participant service benefits become smaller and smaller after the participant is selected. The profit obtained from completing task can be calculated as follow

$$f_i(s_{ij}, d_{ij}) = \sum_{j=1}^n [w_i \cdot \ln(1 + s_{ij})] - \sum_{j=1}^n (P_{ij} \cdot d_{ij}) \quad (3)$$

where system coefficient $w_i > 0$ is determined by the platform for each task type. Apparently, both the platform and participants want to maximize their profits or incomes. Hence, a reasonable task assignment strategy should select participants with low sensing costs and high service benefits, where a tradeoff between the platform and participants must also be made under constraints of the maximum movement distance $r_{\max j}$ and budget of each type of tasks $B = [B_1, B_2 \dots B_m]$. Rationally, the reward should be more than the cost to motivate participants to travel within $r_{\max j}$. Eventually, the task assignment can be formulated as the following optimization problem

$$\max \psi_j(d_{ij}) = \sum_{i=1}^m [(P_i \cdot d_{ij}) - C_{ij}] \quad (4)$$

$$\max f_i(s_{ij}, d_{ij}) = \sum_{j=1}^n [w_i \cdot \ln(1 + s_{ij})] - \sum_{j=1}^n (P_i \cdot d_{ij}) \quad (5)$$

$$\begin{aligned} \text{s.t. } & \sum_{i=1}^m d_{ij} \leq r_j^{\max}, d_{ij} > 0 \\ & \sum_{j=1}^n P_i \cdot d_{ij} \leq B_i \\ & 0 < s_{ij} \leq 1 \end{aligned} \quad (6)$$

4 Service Benefit Evaluation

Service benefit is an important indicator for the platform to estimate the potential profit gained from a certain participant, which often relates to task requirements. In this paper,

task difficulty, task history, sensing capacity, and sensing positivity are employed, from perspectives of tasks, participants, and publishers, to comprehensively evaluate the service benefit of all types of tasks, so as to achieve accurate participant matching and reliable MCS data collection.

4.1. Sensing Positivity

Sensing positivity refers to the motivation of participants in performing sensing tasks, which is a dynamic process. Given the same sensing capacity, a higher sensing positivity signifies a greater contribution to the platform. Interactions between participants and the platform are employed to measure the sensing positivity, where interaction frequency and task performance are two major observable indicators for these interactions. Specifically, performance p is a function of response time Δt and cost C_{ij} . If a participant has a relatively low Δt and C_{ij} , the platform deems his/her performance positive. Due to the restrictive task deadline and movement distance, p decreases with the growing Δt and C_{ij} , and then gradually stabilizes. Therefore, performance p_{ji} of participant u_j in task A_i can be calculated by

$$p_{ji} = \log_2 \left(\frac{1}{C_{ij} \cdot \Delta t} + 1 \right) \quad (7)$$

where $\Delta t = |t_q^i - t_s^i|$, $\Delta t \leq T_d \cdot t_q^i$ is the time participant u_j starts task A_i and obviously $\Delta t = 0$ if t_q^i is equal to task start time t_s^i , which generates the value of maximum performance is 1. Conversely, the minimum performance 0 can be obtained when $\Delta t = T_d$. Besides, when C_{ij} is infinitely large, p_{ji} reaches 0 (i.e., $\log_2 1 = 0$), as bounded by

$$p_{ji} = \begin{cases} 0 & , \Delta t = T_d \\ \log_2 \left(\frac{1}{C_{ij} \cdot \Delta t} + 1 \right) & , 0 < \Delta t < T_d \\ 1 & , \Delta t = 0 \end{cases} \quad (8)$$

Interaction frequency is another important indicator for sensing positivity. Apparently, a high interaction frequency signifies a stable and positive sensing behavior, and thereby the interaction frequency has the equivalent weight with the task performance in sensing positivity. Therefore, the sensing positivity of participant u_j in task A_i can be obtained as

$$\chi_j^i = \frac{f_{jm}}{\sum_{j=1}^n \sum_{i=1}^m f_{ji}} \cdot p_{ji} \quad (9)$$

In Equation (9), $f_j = [f_{j1}, f_{j2}, \dots, f_{jm}]$, $f_{jm} = [1, 2, \dots, h]$ refers to the tasks participated by u_j , and h denotes the latest task. The sensing positivity for different types of tasks is represented by a vector $X_j = [\chi_{1j}, \chi_{2j}, \chi_{3j}, \dots, \chi_{mj}]$

4.2. Task Difficulty

The task difficulty challenges the sensing capacity of participants, and we utilize a difficulty coefficient to measure it in this paper, where a small coefficient signifies a difficult task. However, dynamic MCS tasks are large in number and rich in type. Evaluating the task difficulty in real time will inevitably consume massive computing power of the platform, which is prohibitively expensive. Therefore, we employ an offline method evaluating the completion rate in task history to obtain the difficulty coefficient. Specifically, the completion rate is defined as the ratio of completed

subtasks $\left| A_{competed}^i \right|$ competed to all published subtasks q , $\ell_i = \frac{\left| A_{competed}^i \right|}{q}$ and $\left| A_{competed}^i \right| \leq q$. Note that the completion rate of different types of

tasks is denoted by a vector $\ell = [\ell_1, \ell_2, \dots, \ell_m]$. Intuitively, a high completion rate signifies a simple task. Besides, the completion time must be before the deadline. Here, we exploit a theoretical completion time $\vartheta = [\vartheta_1, \vartheta_2, \dots, \vartheta_m]$ to evaluate the actual completion time $\tilde{\vartheta} = \tilde{\vartheta}_1, \tilde{\vartheta}_2, \dots, \tilde{\vartheta}_m$, and they are calculated by

$$\tilde{\vartheta}_i = \sum_{q=1}^{\left| A_{competed}^i \right|} a_q^i \cdot t_{make-span}^{actual} \quad \text{and} \quad \vartheta_i = q \cdot \sum_{a_q \in A_i} t_{make-span}^{theoretical}$$

where q is the total number of published subtasks. If $\tilde{\vartheta}_i$ is within ϑ_i , this task can be completed. Eventually, the difficulty coefficient can be obtained as

$$D_i = \begin{cases} \ell_i \times \frac{(\vartheta_i - \tilde{\vartheta}_i)}{\max(\vartheta_i - \tilde{\vartheta}_i)}, & \vartheta_i \geq \tilde{\vartheta}_i \\ \ell_i \times \left(1 - \frac{(\tilde{\vartheta}_i - \vartheta_i)}{\max(\tilde{\vartheta}_i - \vartheta_i)} \right), & \vartheta_i < \tilde{\vartheta}_i \end{cases}, \vartheta_i \leq T_d \quad (10)$$

$$\mathfrak{S}_{ij} = \begin{cases} \log_2 (|\kappa_{ij} - D_i| + 1), & \vartheta_i \geq \tilde{\vartheta}_i \\ \log_2 (|\kappa_{ij} - D_i| + 1), & \vartheta_i < \tilde{\vartheta}_i \end{cases}, \vartheta_i \leq T_d \quad (11)$$

Relative difficulty coefficients are further denoted by matrix $C_{n \times m}$, as shown in Equation (12), where rows represent participants and columns indicate task

$$C_{n \times m} = \begin{bmatrix} \mathfrak{S}_{1,1} & \mathfrak{S}_{1,2} & \dots & \mathfrak{S}_{1,m} \\ \mathfrak{S}_{2,1} & \mathfrak{S}_{2,2} & \dots & \mathfrak{S}_{2,m} \\ \dots & \dots & \dots & \dots \\ \mathfrak{S}_{n,1} & \mathfrak{S}_{n,2} & \dots & \mathfrak{S}_{n,m} \end{bmatrix} \quad (12)$$

4.3. Task History

Generally, subjective feedback from task publishers is an effective benchmark for the credibility of data submitted by participants. However, due to insufficient labeled MCS data, it is challenging to objectively evaluate the data credibility. We exploit historical records of participants including ID, collected data format v_j , $v_j \in v$ and reward $d_{<j, <j} = P_i \cdot d_{ij}$, as denoted by $Z_j = \#$, v_j , $<j$, to evaluate data credibility. Collected data format v_j is compared with task requirements v defined by T_{doc} , and a small gap signifies the complete data, which can be indicated by $I_j = \frac{\|v - v_j\|}{\max\|v - v_j\|}$. Besides, the more data formats the platform receives, the higher data credibility a task can obtain. Generally, the value of data is defined as a quotient of frequency and the residual of v_j in the task history of publisher R_i . The value of the historical data is estimated by linear regression model $Y = X\omega + \delta$, where Y and δ are both $|U|$ dimension vector, $|U|$ denotes the number of participants, X is a $|U| \times |Z_i|$ matrix, and ω is a $|Z_i|$ dimension vector. Therefore, the residual between the actual data value and the estimated is $\hat{\delta} = Y - Y^{\wedge} = (I - M) Y$, where $M = X X^T X^{-1} X^T$ is a hat matrix and a small residual indicates high data credibility. Consequently, the data credibility of participant u_j for task A_i can be obtained as $reliable_{ij} = I_j \cdot f_j$ value $k \hat{\delta}_k$. Moreover, a publisher gives a high score to participants requiring relatively low task payments, and the score given to participants can be obtained as follow

$$n_{ij} = \frac{reliable_{ij}}{\mathfrak{R}_j} \quad (13)$$

We further adopt the logarithm to normalize the score, as shown in the follow

$$n_{ij} = \log_2 \left(1 + \frac{reliable_{ij}}{\mathfrak{R}_{ij}} \right) \quad (14)$$

Then, the score matrix $N_{n \times m}$ can thus be obtained as follows

$$N = \begin{bmatrix} n_{11} & n_{12} & n_{13} & \dots & n_{1m} \\ n_{21} & n_{22} & n_{23} & \dots & n_{2m} \\ n_{31} & n_{32} & n_{33} & \dots & n_{3m} \\ \dots & \dots & \dots & \dots & \dots \\ n_{n1} & n_{n2} & n_{n3} & \dots & n_{nm} \end{bmatrix} \quad (15)$$

where rows represent task publishers and columns represent participants. 4.4. Service Benefit Mathematically, the service benefit of participants is a function of task score n_{ij} , relative task difficulty \mathfrak{R}_{ij} , and sensing positivity χ_{ij} , where χ_{ij} serves as the weight of the service benefit indicating the motivation of participants. In addition, the service benefit grows monotonically with the increasing task score and difficulty. As marginal benefits of submitted data gradually decrease, the growth rate of service benefits drops and stabilizes. Thus, the service benefit of u_j to A_j can be formulated through the following inverse trigonometric function

$$s_{ij} = \frac{\chi_j^i}{\pi} \times \arctan [n_{ij} \cdot \mathfrak{R}_{ij}] + \frac{1}{2} \quad (16)$$

However, the evaluation of service benefits depends on task history, which is inapplicable to new participants. Therefore, we propose to set the default value of service benefits to 0.5, indicating an uncertain service benefit for strange participants. Besides, 0.5 also serves as a threshold to distinguish participants with low service benefits. Based on their task histories, we can reformulate the service benefit of participants as

$$s_{ij} = \begin{cases} \frac{\chi_j^i}{\pi} \times \arctan [n_{ij} \cdot \mathfrak{R}_{ij}] + \frac{1}{2}, & \text{task history} \\ 0.5 & , \text{no task history} \end{cases} \quad (17)$$

Similarly, service benefits of participants for different types of tasks can also be denoted by matrix $S_{n \times m}$.

5. Service Benefit Aware Multi-Task Assignment

Technically, the optimization goal of MCS task assignment is to select participants with high service benefits and low costs, so as to balance the interests of participants and the platform, given constraints of movement distance and budget. To solve this optimization problem, we first cluster users (i.e., task candidates) according to their task preferences, and then exploit a gradient descent algorithm to find the optimal participants in each cluster.

5.1. User Clustering Based on Task Preference

In MCS scenarios with massive users, the matching accuracy of optimization algorithms always suffers from the large search range of task candidates. Therefore, we propose to employ the similarity among task preferences to cluster task candidates. Specifically, task preferences indicate the interest of users in certain tasks, which can be reflected by task

acceptance rate and task performance. The task acceptance rate is defined as the

$$p_{acc}^{jm} = \frac{|SA_i|_{u_j}}{\sum_{u_j \in W_{i=1}} \sum_k |SA_i|_{u_j}},$$

proportion of tasks submitted by user u_j to the total number of tasks

which further serves as the weight of the task preference. Intuitively, the acceptance rate

$$\frac{|SA_i|_{u_j}}{\sum_{u_j \in W_{i=1}} \sum_k |SA_i|_{u_j}},$$

of participant, calculated by $p_{acc}^{jm} =$ which further serves as the weight of the task preference. Intuitively, the acceptance rate of participant u_j for different types of tasks can be denoted by vector $r P^{ji}_{acc} = [p^{j1}_{acc}, p^{j2}_{acc} \dots \dots p^{jm}_{acc}]$ The task preferences of users $p_j = [p^{j1}, p^{j2} \dots \dots p^{jm}]$ are also perceived by the platform, which can be calculated similarly with Equation (8). Therefore, task preferences can be denoted by the product of the task acceptance rate and task performance (i.e., $h_{ij} = p^{ji}_{acc} \times p^{ji}$), and its matrix holds as

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} & \dots & h_{1m} \\ h_{21} & h_{22} & h_{23} & \dots & h_{2m} \\ \dots & \dots & \dots & \dots & \dots \\ h_{n1} & h_{n2} & h_{n3} & \dots & h_{nm} \end{bmatrix} \quad (18)$$

where rows represent users and columns represent tasks. The number of MCS clusters depends on the number of published tasks in each time interval, which varies dynamically with the task preference. Note that a user may be interested in multiple tasks and therefore belongs to more than one cluster, which makes Fuzzy C-Means (FCM) algorithm a perfect clustering method for this scenario. In terms of the task preference defined above, we employ cosine similarity to replace the Euclidean distance in standard FCM and modify it into similarity FCM (SFCM). The cosine similarity of task preferences indicating the preference similarity between cluster center ok and user uj in SFCM can be calculated by

$$d_{kj} = 1 - \cos(h_k, h_j) = 1 - \frac{\sum_{i=1}^m h_{ji} \times h_{ki}}{\sqrt{\sum_{i=1}^m h_{ji}^2} \sqrt{\sum_{i=1}^m h_{ki}^2}} \quad (19)$$

In FCM, the fuzzy weighted exponent m is commonly employed to determine the fuzzy degree of clustering results, and its optimal value is usually set to $1.5 \leq m \leq 2.5$. We take $m = 2$ and the objective function of SFCM can be obtained as follows

$$J(U, O) = \sum_{j=1}^n \sum_{k=1}^{\Lambda} (\mu_{kj})^2 (d_{kj})^2$$

$$s.t. \begin{cases} 0 \leq \mu_{kj} \leq 1, k \in [1, \Lambda], j \in [1, n] \\ \sum_{k=1}^{\Lambda} \mu_{kj} = 1, j \in [1, n] \\ 0 < \sum_{j=1}^n \mu_{kj} \leq n, k \in [1, \Lambda] \end{cases} \quad (20)$$

In Equation (20), μ_{kj} represents the membership degree of user u_j to cluster O_k . The membership matrix can then be denoted by $U_{\Lambda \times n}$ and the cluster center matrix is O , which can be calculated in the following

$$o_k = \frac{\sum_{j=1}^n (\mu_{kj})^2 h_{ji}}{\sum_{j=1}^n (\mu_{kj})^2}, k \in [1, \Lambda] \quad (21)$$

$$\mu_{kj} = \frac{1}{\sum_{i=1}^{\Lambda} \left\{ \frac{d_{ki}}{d_{ji}} \right\}^2}, k \in [1, \Lambda], j \in [1, n] \quad (22)$$

We set the iteration times to l and the stop parameter to ξ , respectively. Given the user preference matrix H , SFCM randomly generates an initial membership matrix U_0 and calculates Λ initial cluster centers $o_k, k \in (1, 2, \dots, \Lambda)$. According to cluster center matrix O , both cosine similarity d_{kj} and membership matrix $U_{\Lambda \times n}$ can be obtained. For instance, if $d_{kj} = 0$, membership degree of u_j to O_k is 1. Finally, the iteration is stopped, if $J(l+1) - J(l) \leq \xi$, to generate clustering results $U_{\Lambda \times k}$ and obtain cluster center matrix O . Otherwise, iterations continue to update $U_{\Lambda \times k}$ and O until reaching iteration times l or stop parameter ξ . Generally, FCM constructs clusters according to the membership matrix (i.e., $O_k = \{k | \mu_{kj} = \max_i u_{ij}, 1 \leq j \leq \Lambda\}, u_j \in O_k$). However, users clustered by FCM can only belong to one cluster according to her/his highest task preference, which is against the intuition that users with similar preferences for several tasks may simultaneously belong to multiple clusters. Therefore, we define clustering threshold Θ to establish these characteristic overlapping clusters. Specifically, the maximum membership value of a user in the cluster is compared with the other memberships value that he/she belongs to, then all comparison values are sorted, the largest comparison value are got among them as the threshold, which calculated by $\Theta = \arg \max_{E_j} \bigcup_{j=1}^{|U_k|} E_j, E_j = \bigcup_{i=1}^m \min \left\{ \mu_{ij}^{\max} - \mu_{-ij} \right\}$.

The true label is obtained by the maximum average preference value among different types of tasks in a cluster; there holds

$$\bar{H}_i = \frac{\sum_{j=1}^{|U_k|} h_{ij}}{|U_k|} \quad (23)$$

where $|U_k|$ is the total number of users in cluster O_k .

5.2. Optimization Problem Based on Lagrange Duality

5.2.1. Problem Reformulation

Since the task price per meter is fixed, the income of participants can be maximized by reducing the movement distances, whereas the platform maximizes its profit by selecting participants with high service benefits. Therefore, the optimization problems in Equations (4)–(6) can be rewritten as follows

$$\begin{aligned}
 & \max \sum_{i=1}^m \varphi_i - \sum_{j=1}^n C_j \\
 & \text{s.t.} \sum_{i=1}^m d_{ij} \leq r_j^{\max}, d_{ij} > 0, j = 1, 2, \dots, n \\
 & \sum_{j=1}^n P_i \cdot d_{ij} \leq B_i, i = 1, 2, \dots, m \\
 & 0 < s_{ij} \leq 1
 \end{aligned} \tag{24}$$

$$\varphi_i = \sum_{j=1}^n w_i \cdot \ln(1 + s_{ij}) \text{ and } C_j = \sum_{i=1}^m (a_{ij} \cdot d_{ij}^2 + b_{ij} \cdot d_{ij} + c_{ij})$$

In Equation (24), φ_i and C_j represent the service benefits obtained by the platform and the costs consumed by participants, respectively.

5.2.2. Lagrange Duality

Equation (24) shows that the objective function is convex with respect to d_{ij} and s_{ij} . Hence, the Lagrange multiplier can be employed to solve this unconstrained dual problem; there hold

$$\begin{aligned}
 L(d_{ij}, \alpha_{ij}, \beta_{ij}) &= \sum_{i=1}^m \varphi_i - \sum_{j=1}^n C_j + \sum_{j=1}^n \alpha_{ij} \cdot \left(\sum_{i=1}^m d_{ij} - r_j^{\max} \right) + \sum_{i=1}^m \beta_{ij} \cdot \left(\sum_{j=1}^n P_i \cdot d_{ij} - B_i \right) \\
 &= \sum_{i=1}^m \left[\varphi_i + \beta_{ij} \cdot \left(\sum_{j=1}^n P_i \cdot d_{ij} - B_i \right) \right] + \sum_{j=1}^n \left[\alpha_{ij} \cdot \left(\sum_{i=1}^m d_{ij} - r_j^{\max} \right) - C_j \right]
 \end{aligned} \tag{25}$$

In Equation (25), the Lagrange multiplier is denoted by matrix $\beta_{m \times n} = [\beta_{1,*}, \beta_{2,*}, \dots, \beta_{m,*}]$, $\alpha_{n \times m} = [\alpha_{1,*}, \alpha_{2,*}, \dots, \alpha_{n,*}]$ and $\alpha_{ij} \geq 0$. Since the service benefit of participants is already evaluated, the dual problem can be defined by

$$\begin{aligned}
 \min_{\alpha_{ij} \geq 0, \beta_{ij}} D(\alpha, \beta) &= \min_{\alpha_{ij} \geq 0, \beta_{ij}} \max_{d_{ij} > 0} L(d_{ij}, \alpha_{ij}, \beta_{ij}) \\
 &= \min_{\alpha_{ij} \geq 0, \beta_{ij}} \max_{d_{ij} > 0} \sum_{i=1}^m \Phi_i(\beta_{ij}) + \sum_{j=1}^n \Psi_j(\alpha_{ij})
 \end{aligned} \tag{26}$$

In Equation (26),

$$\Phi_i(\beta_{ij}) = \max_{d_{ij} \geq 0} \varphi_i + \beta_{ij} \cdot \left(\sum_{j=1}^n P_i \cdot d_{ij} - B_i \right) \text{ and } \Psi_j(\alpha_{ij}) = \max_{d_{ij} \geq 0} \alpha_{ij} \cdot \left(\sum_{i=1}^m d_{ij} - r_j^{\max} \right) - C_j.$$

Because the original objective function is convex, the strong duality must satisfy the Slater condition to generate the optimal solution for this dual problem.

5.2.3. Optimization Algorithm We employ a gradient descent algorithm to iteratively solve the dual problem. The variables of the dual problem can be updated as follows

$$\begin{aligned}
 \langle \alpha_{ij}^{l+1}, \beta_{ij}^{l+1} \rangle &= \left[\langle \alpha_{ij}^l, \beta_{ij}^l \rangle - \lambda \cdot \left\langle \frac{\partial D}{\partial \alpha_{ij}}, \frac{\partial D}{\partial \beta_{ij}} \right\rangle \right]^+ \\
 &= \left[\langle \alpha_{ij}^l, \beta_{ij}^l \rangle - \lambda \left\langle \left(d_{ij}^l - r_j^{\max} \right), \left(P_i \cdot d_{ij}^l - B_i \right) \right\rangle \right]^+
 \end{aligned} \tag{27}$$

In Equation (27), d_{ij}^l is the variable of the original optimization problem in the l th iteration, α_{ij}^l and β_{ij}^l are the variables of the dual problem in the l th iteration, and λ is the learning step size. Participants with the best service benefits and optimal movement distances can be obtained iteratively by the platform. First, in the iteration of service benefits, participants with the best benefits in the l th iteration can be obtained. Then, in the gradient descent algorithm, dual variables α_{ij}^l and β_{ij}^l of the l th iteration are obtained. Finally, s_{ij}^l , α_{ij}^l and β_{ij}^l are all set for Equation (28) to generate the optimal

movement distance. The iteration process does not stop until convergence conditions are met. Algorithms 1 and 2 are updated as Equations (28) and (29),

$$\forall u_j \in U, d_{ij}^l = \arg \max \varphi_i + \beta_{ij}^l \cdot \left(\sum_{j=1}^n P_i \cdot d_{ij} - B_i \right) + \alpha_{ij}^l \cdot \left(\sum_{i=1}^m d_{ij} - r_j^{\max} \right) - C_j \quad (28)$$

$$\forall A_i \in A, s_{ij}^l = \arg \max \varphi_i \quad (29)$$

Specifically, the complexity of Algorithm 1 is $O(n)$, where n is number of task candidate. The complexity of Algorithm 2 is $O(m)$, where m is the number of task cluster. The complexity of overall assignment strategy is $O(m \times n)$. When the best participants are selected for each type of task, the platform pays their task reward, updates their service benefits, and exploits scores from task publishers for the next round participant selection.

Algorithm 1 Service benefits.

Input: potential participants set U ; service benefit matrix S ; Task set A

Output: the optimal participants W for Task A_i ; Profit $f_i(s_{ij}, d_{ij})$

Initialize

$W = \phi$, the number of iterations $l = 0$

Select participant u_0 randomly, $W \leftarrow u_0$

Receive d_{ij}^l from user

Calculate the profit of each task of u_0 through Equation (5)

while $|f_i^{l+1}(W) - f_i^l(W)| \leq 10 \times e^{-k}$ or $l \neq 0$ **do**

if $s_{ij} = \arg \max \varphi_i$ **then**

 Select participant u_j and corresponding s_{ij}

 Calculate the profit of the platform through Equation (5))

return $\langle s_{ij}, f_i^l(s_{ij}, d_{ij}) \rangle$

break

else

$l = l - 1$

 update Equation (29)

 return to Line 5

end if

end while

return $\langle s_{ij}, f_l(W) \rangle$

Algorithm 2 Iterative of d_{ij} progress.

```

Initialization
for  $l = 0, 1, 2, 3, \dots$  do
    Receive  $s_{ij}^l$  from platform
    Update Equation (28)
    Compute the new value of  $\alpha_{ij}^{l+1}$  and  $\beta_{ij}^{l+1}$  using Equation (27)
    if  $|\alpha_{ij}^{l+1} - \alpha_{ij}^l| \geq \rho$  and  $|\beta_{ij}^{l+1} - \beta_{ij}^l| \geq \rho$ , where  $\rho$  is a tunable little real number then
        Return to Line 3
    else
        return  $\langle d_{ij}^l, \psi_j(d_{ij}) \rangle$ 
        break
    end if
end for
    
```

6. Experiment

Gowalla, employed in this study to validate the proposed SBAMA, is a location-based real world social network dataset that allows users to share their information, including ID, access time, longitude, latitude, and location tags. The dataset collected all public check-in data between February 2009 and October 2010. There are 19,6591 nodes and 950,327 edges in Gowalla. Gowalla is mainly used to study human mobility. Specifically, 500 locations and 1000 users were extracted from Gowalla as task locations and candidates, respectively. Subsequently, these 1000 task candidates were cluster into five groups, where each group maintains a task preference matrix and a corresponding service benefit matrix, containing five types of tasks. In addition, SFCEM clustering algorithm and optimization algorithm in SBAMA were compared with original FCM algorithm and greedy algorithm in Dynamic Trust-Based Recruitment Framework (DTRF) on MATLAB platform, respectively. Simulation parameters are given in Table 1.

Table1: Parameter values

Parameter	Value
target region (km ²)	600 × 50
types of task	5
number of unit tasks for each task	[20, 200]
participant service benefit value	(0, 1)
user preference value inherent sensing cost	
c_{ij}	(1, 4) (0, 1)
a_{ij}	(0, 1)
b_{ij}	(0, 1)
c_{ij}	(1, 2)
maximum movement distance $r_{\max j}$ (km)	10
task effective time (min)	[20, 60]
task price per meter p_{ij} (\$/m)	(0.2, 0.5)
number of candidate users	200
ξ	10–5
w_i	(100, 300)
B_i	[2000, 5000]

6.1. Advantages of SFCM

The objective function iteration and clustering accuracy of SFCM were compared with those of FCM to verify the effectiveness of SFCM. Objectively, both FCM and SFCM adopt the same initial membership matrix and the simulation was repeated 100 times, where seven tests were randomly selected for observation. The iteration times needed by FCM and SFCM for objective function convergence are shown in Figure 2. Compared with FCM, SFCM requires a stably lower number of iterations around 30. In addition, SFCM converges quickly and has significantly short clustering time. Figure 3 illustrates the iteration of their objective function values, where the initial value of SFCM is notably much smaller than that of FCM, because the Euclidean distance in FCM is replaced by cosine similarity of SFCM to reduce the membership value.

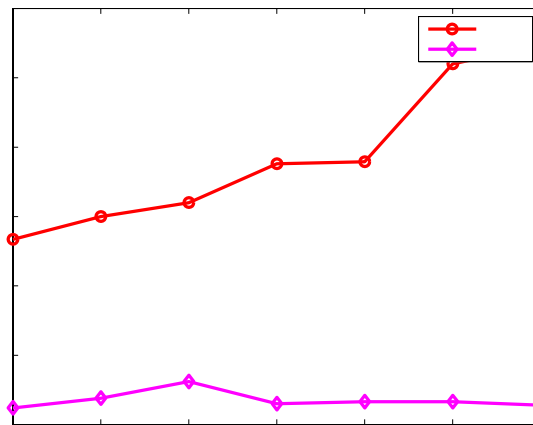


Figure 2. Comparison of the number of iterations.

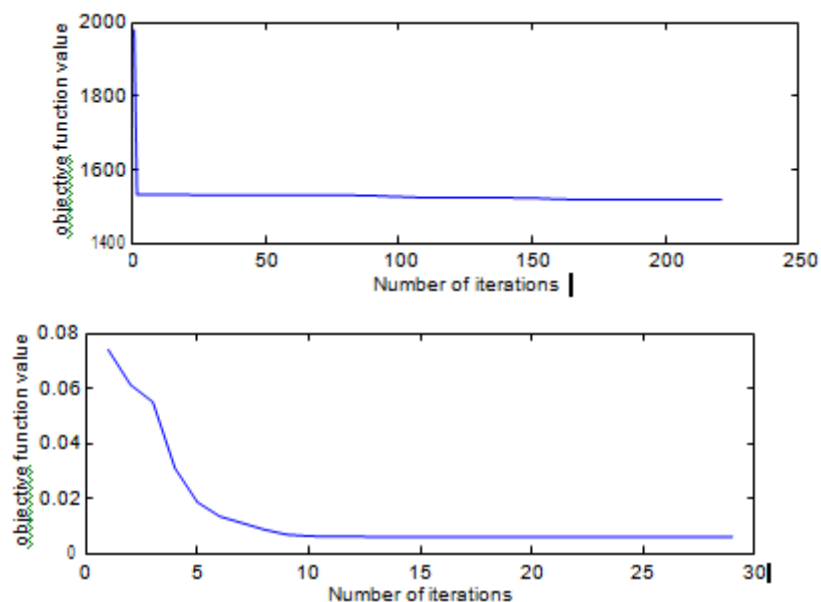


Figure 3. Comparison of the iterative process.

The clustering accuracy of FCM and SFCM, given the maximum membership value, is shown in Figure 4, which is measured based on the original dataset with labels. The clustering accuracy of SFCM is generally higher than 95%, whereas the worst case of FCM is only 74.5%. Similarly, given the maximum membership value, randomly selected

clustering results of FCM and SFCM are shown in Figure 5, where SFCM has a significantly better clustering result. Figure 6 shows the final membership matrix value of users from a random test. Cluster labels can be determined by Equation. For example, the cluster for Task A1 almost includes Users 40–80. However, according to their membership matrices, Users 34, 84, 100, 103, and 156 have similar membership values for different types of tasks. As shown in Table 2, the membership values of User 34 for Tasks A3 and A5 only differ by approximately 0.074. Besides, Users 103 and 156 have membership differences only within 0.1 for Tasks A2/A4 and Tasks A2/A5, respectively. Therefore, the task preference threshold is set to 0.1 for overlapping clustering and the clustering result of SFCM based on this threshold is shown in Figure 7. Compared with Figure 5, clusters overlap and User 34 belongs to clusters of Task A3 and A5 simultaneously, which is more practical for real world MCS scenarios. In short, SFCM with membership threshold can cluster users with similar task preferences, which is an effective underpinning for the subsequent optimization problem.

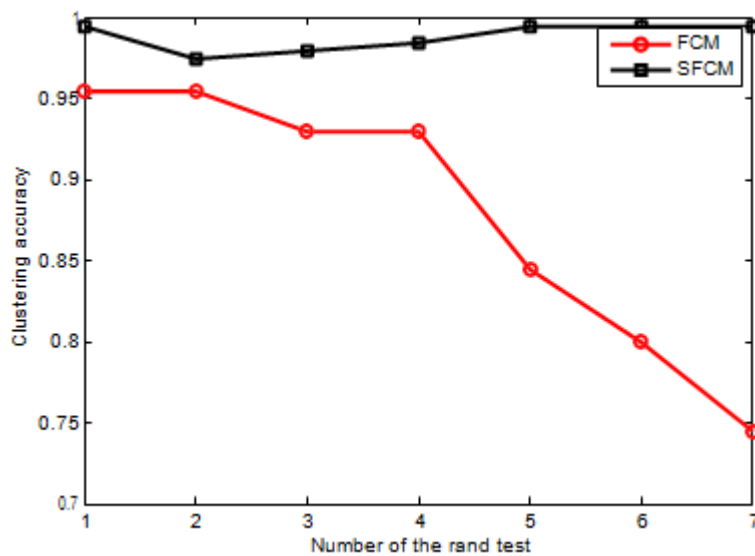
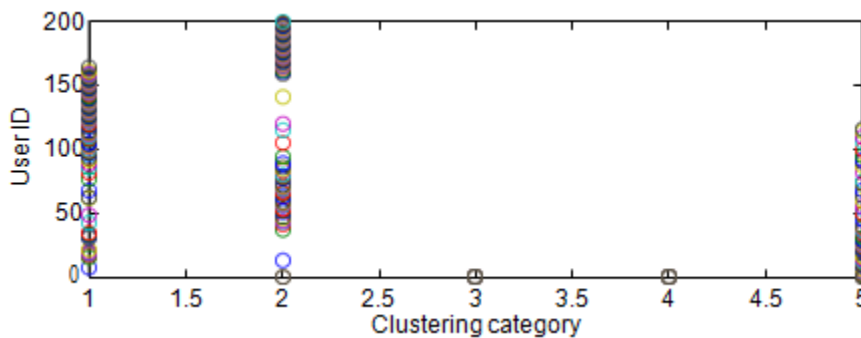


Figure 4. The accuracy of clustering.



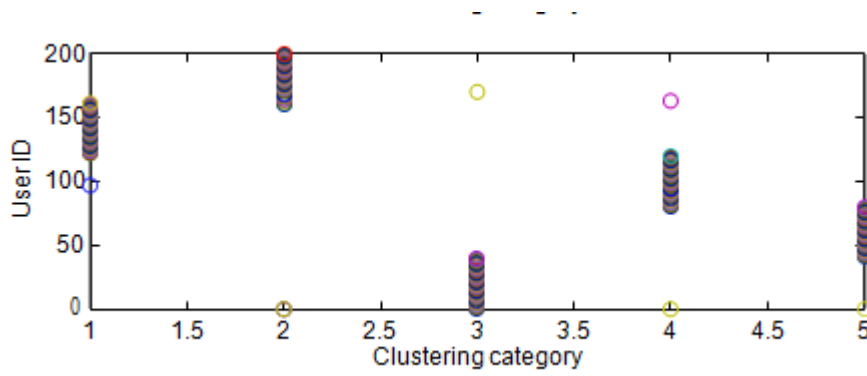


Figure 5. Comparison of user clustering.

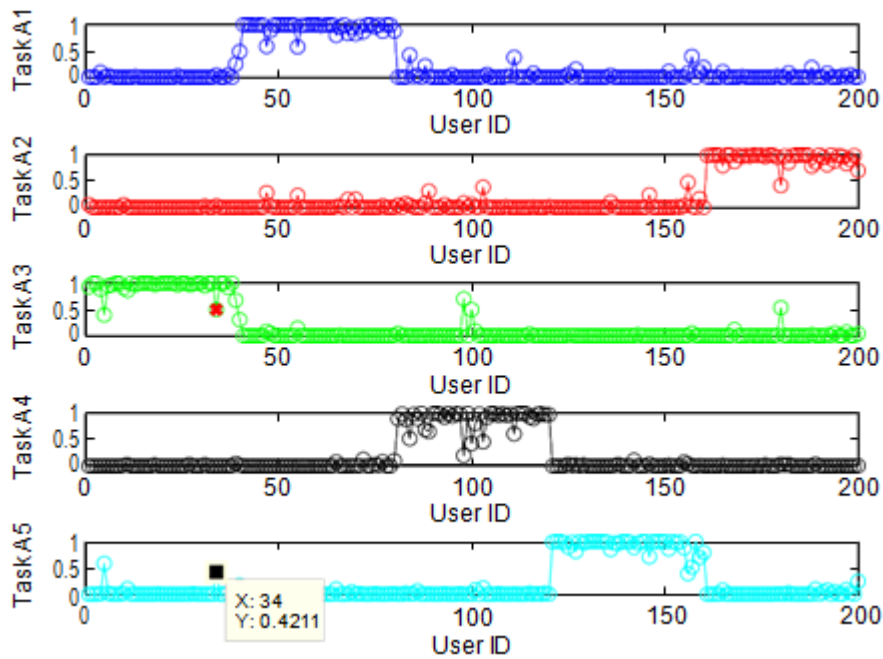


Figure 6. Membership matrix values of SFCM.

Table 2. Similar membership values of randomly tested users.

Task User	34	84	100	103	156
A1	0.046355202	0.423835317	0.004346063	0.021614770	0.087140617
A2	0.034177025	0.046220128	0.066147715	0.388445033	0.479426570
A3	0.494624410	0.003243961	0.497748453	0.009119740	0.002899209
A4	0.003762370	0.517018454	0.425430680	0.458955766	0.039708579
A5	0.421080991	0.009682137	0.006327087	0.121864688	0.390825023

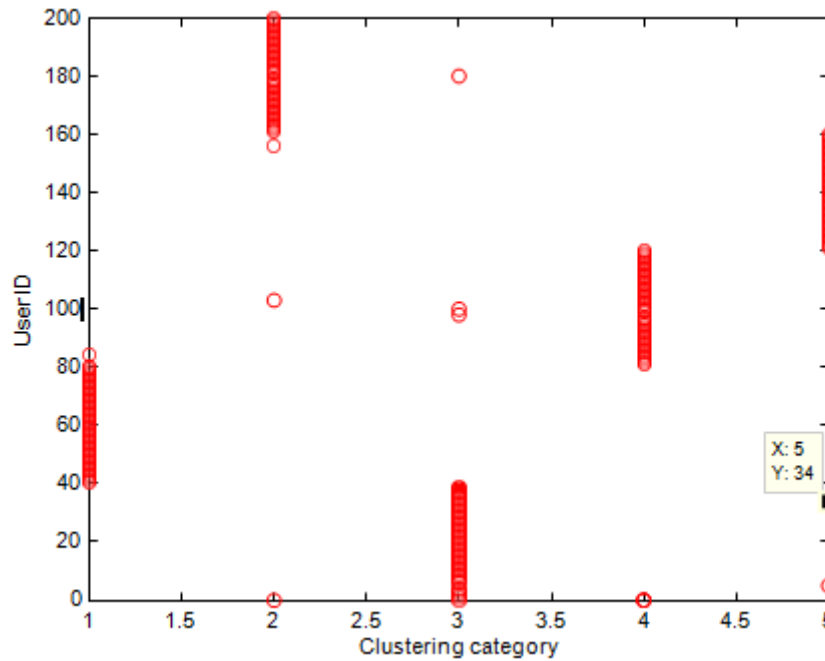


Figure 7. Result of SFCM cluster based on threshold.

6.2. Analysis of Optimization Algorithm

Figure 8 shows how the platform profit gained from each type of task varies with the number of iterations. It is observed that the platform profit converges to the optimal value when the number of iterations reaches about 45, which implies the platform can stably match appropriate participants to tasks. Besides, the platform profit increases as the average service benefit increases. In Figure 9, the impact of the number of iterations on the participant income is depicted. For a fixed task price P_i , the participant income first grows sharply as the number of iterations increases and then tends to be stable. The income of Participant 3 is significantly higher than those of others. This is because his/her task is more difficult to be performed and requires a stronger sensing capacity, which thereby receives a higher payment from the platform. In Figures 8 and 9, the fast iteration convergence of the proposed gradient descent algorithm for achieving the best task participants is validated.

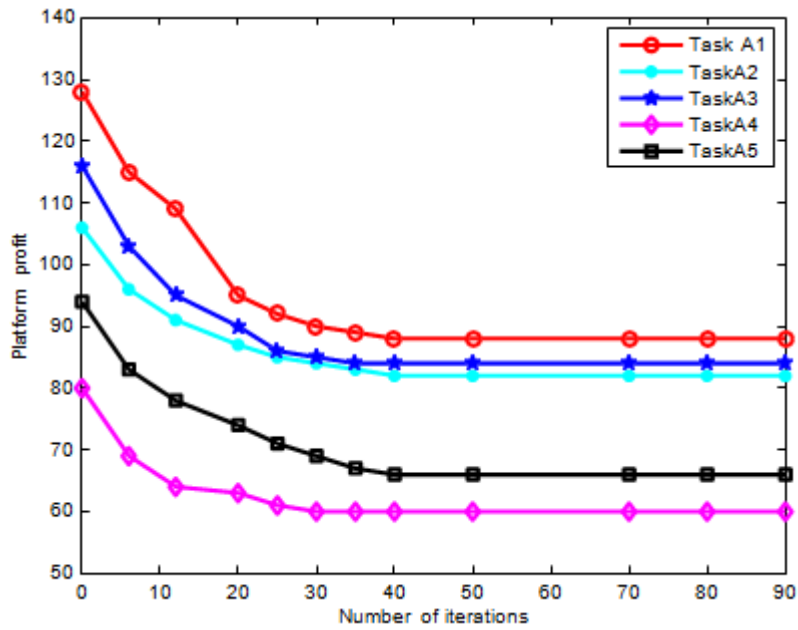


Figure 8. Changes in platform profit.

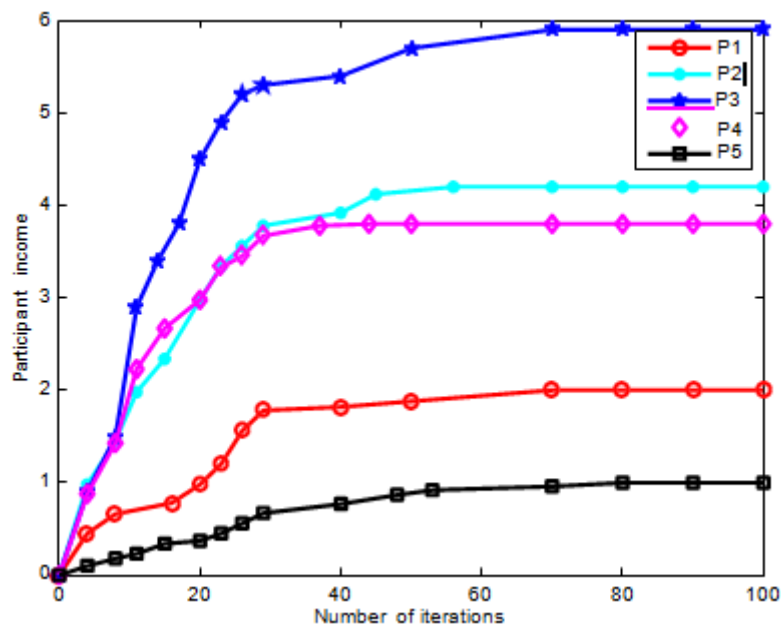


Figure 9. Changes in participant income.

In Figure10, the task completion rates of DTRF and SBAMA are compared. In addition to sensing quality considered by DTRF, the proposed SBAMA also takes service benefits of participants, task preferences and real-time feedbacks from task publishers. Hence, SBAMA acquires 8% higher task completion rate.

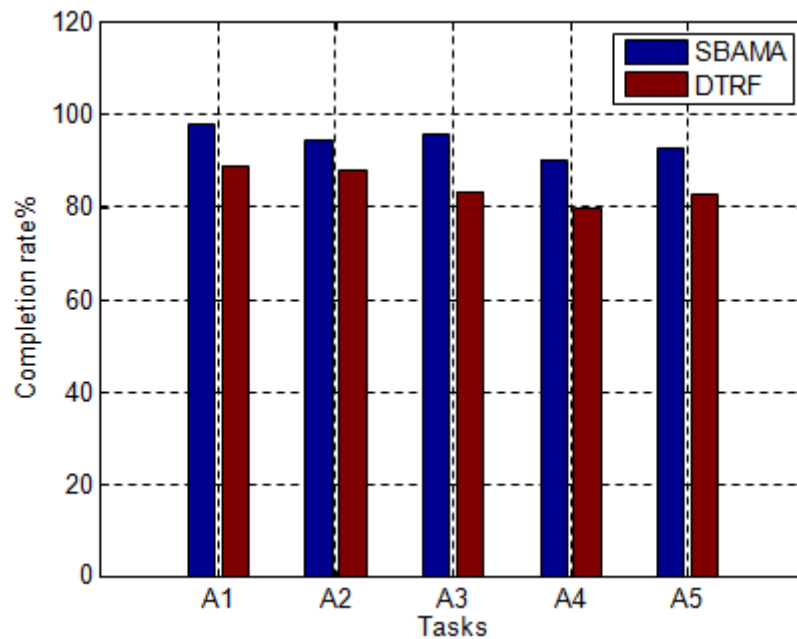


Figure 10. Comparison of task completion rates for different tasks.

In short, through narrowing the search range of task candidates, SBAMA effectively improves the matching accuracy of tasks assignment with fast algorithm convergence.

7. Conclusions

In this paper, we propose the SBAMA to quickly and accurately match MCS tasks with the most appropriate participants to improve the task completion rate and data credibility. Firstly, the service benefit of participants is modeled based on their task difficulty, task history, sensing capacity and sensing positivity to improve the accuracy of task assignment. Then, task candidates are clustered according to their task preference to narrow the search range. Finally, the gradient descent algorithm is designed to select the optimal participants in each cluster. Simulation results verify that the proposed SBAMA can quickly find the most appropriate participants to meet the requirements of multiple concurrent types of tasks under a massive user scenario, for example, crowded road condition monitoring. Although the proposed SBAMA can be applied to the scenario of massive users and numerous concurrent tasks, the strategy still has some limitations for the scenario where participants are sparse, which can lead to a low matching accuracy between the task and the participant. In the future, we will focus on the sparse participant scenario and study the associated task assignment strategy.

REFERENCES

- [1]. Z. Zhang and L. Wang, "Social tie-driven content priority scheme for D2D communications," *Inf. Sci.*, vol. 480, pp. 160_173, Apr. 2019.
- [2]. D.Wu, B. Liu, Q. Yang, and R. Wang, "Social-aware cooperative caching mechanism in mobile social networks," *J. Netw. Comput. Appl.*, vol. 149, Jan. 2020.
- [3]. R.Wang, J. Yan, D. Wu, H. Wang, and Q. Yang, "Knowledge-centric edge computing based on virtualized D2D communication systems," *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 32_38, May 2018.
- [4]. I. Butun, S. D. Morgera, and R. Sankar, "A survey of intrusion detection systems in wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 266_282, 1st Quart., 2014.
- [5]. X. Bai, L. Liu, M. Cao, J. Panneerselvam, Q. Sun, and H. Wang, "Collaborative actuation of wireless sensor and actuator networks for the agriculture industry," *IEEE Access*, vol. 5, pp. 13286_13296, Jul. 2017.
- [6]. R. Wang, H. Liu, H. Wang, Q. Yang, and D. Wu, "Distributed security architecture based on blockchain for connected health: Architecture, challenges, and approaches," *IEEE Wireless Commun.*, vol. 26, no. 6, pp. 30_36, Dec. 2019.

- [7]. Q. Wang, J. Zheng, H. Xu, B. Xu, and R. Chen, "Roadside magnetic sensor system for vehicle detection in urban environments," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 5, pp. 1365_1374, May 2018.
- [8]. D. Wu, J. He, H. Wang, C. Wang, and R. Wang, "A hierarchical packet forwarding mechanism for energy harvesting wireless sensor networks," *IEEE Commun. Mag.*, vol. 53, no. 8, pp. 92_98, Aug. 2015.
- [9]. D. Wu, Z. Zhang, S. Wu, J. Yang, and R. Wang, "Biologically inspired resource allocation for network slices in 5G-enabled Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9266_9279, Dec. 2019.
- [10]. D.Wu, H. Shi, H. Wang, R. Wang, and H. Fang, "A feature-based learning system for Internet of Things applications," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1928_1937, Apr. 2019.
- [11]. Z. Miao, Y. Wang, and Z. Han, "A supplier-rm-buyer framework for computation and content resource assignment in wireless virtual networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 8, pp. 4116_4128, Aug. 2019.
- [12]. Z. Li, Y. Jiang, Y. Gao, L. Sang, and D. Yang, "On buffer-constrained throughput of a wireless-powered communication system," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 2, pp. 283_297, Feb. 2019.
- [13]. X. Lu, P. Wang, D. Niyato, D. I. Kim, and Z. Han, "Wireless networks with RF energy harvesting: A contemporary survey," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 2, pp. 757_789, 2nd Quart., 2015.
- [14]. C. Wang, J. Li, Y. Yang, and F. Ye, "Combining solar energy harvesting with wireless charging for hybrid wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 3, pp. 560_576, Mar. 2018.
- [15]. P. Zhang, X. Kang, D. Wu, and R. Wang, "High-accuracy entity state prediction method based on deep belief network toward IoT search," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 492_495, Apr. 2019.
- [16]. J. Yang, H. Zhang, Y. Ling, C. Pan, and W. Sun, "Task allocation for wireless sensor network using modified binary particle swarm optimization," *IEEE Sensors J.*, vol. 14, no. 3, pp. 882_892, Mar. 2014.
- [17]. H.-S. Kim, H. Kim, J. Paek, and S. Bahk, "Load balancing under heavy traffic in RPL routing protocol for low power and lossy networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 4, pp. 964_979, Apr. 2017.
- [18]. H. Chen, X. Li, and F. Zhao, "A reinforcement learning-based sleep scheduling algorithm for desired area coverage in solar-powered wireless sensor networks," *IEEE Sensors J.*, vol. 16, no. 8, pp. 2763_2774, Apr. 2016.