A Novel Task Assignment Adjustment Method in Spatial-Temporal Crowdsourcing

Bingyi Sun[1,3], Jiaxu Cui[2,3], Hongtao Bai[1,3], and Yonggang Zhang[2,3](✉)

[1] Public Computer Education and Research Center, Jilin University, Changchun, China
{bysun,baiht}@jlu.edu.cn
[2] College of Computer Science and Technology, Jilin University, Changchun, China
{cjx,zhangyg}@jlu.edu.cn
[3] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, People's Republic of China

Abstract. With the rapid development of mobile networks and the ubiquity of mobile devices, spatial-temporal crowdsourcing, which refers to assigning spatial-temporal tasks to moving workers, has drawn increasing attention. Many researchers aim at various task assignment methods in spatial-temporal crowdsourcing. However, unexpected situations reduce the reliability of the original assignment, such as the absence of reserved workers. To solve the problem, we propose a novel task assignment adjustment method in spatial-temporal crowdsourcing. We design a multi-objective optimization algorithm to minimize the adjustment and maximize the total matching degree in the reassignment process. The experimental results on three real data sets show that the proposed method can improve the total matching degree by about 10% while minimizing the adjustment compared with the baselines.

Keywords: Task assignment · Spatial-temporal crowdsourcing · Multi-objective optimization

1 Introduction

With the increasing popularity of intelligent mobile devices and the rapid development of wireless network technology, Spatial-temporal Crowdsourcing (SC), as a new crowdsourcing model, came into being. Different from traditional crowdsourcing, tasks of SC are spatial-temporal dependent, which means requesters submit tasks with specified execution locations and sensing period. As the smart device carriers, workers need physically move to tasks' locations and accomplish them within the task sensing period.

In spatial-temporal crowdsourcing, task assignment is a necessary research issue, which directly relates to the platform revenue and user experience. However, almost all of the solutions to solve the critical issue follow the rule of

A. Zaslavsky et al. (Eds.): MobiQuitous 2023, LNICST 594, pp. 105–124, 2024.
https://doi.org/10.1007/978-3-031-63992-0_7

immutability [21–23], i.e., after the platform formulates the task allocation scheme, the scheme will not change [11]. Actually, the assignment schemes need to be adjusted dynamically, which is contrary to the rule of immutability. For example, a worker with higher matching degree logs in the task assignment platform at a certain moment, but the platform has recruited enough workers to the spatial-temporal task. According to the rule of immutability, even though the worker has a high matching degree with the spatial-temporal task, he cannot get the opportunity to execute it [19].

Some researchers try to predict the worker future state, and assign the spatial-temporal tasks according to the prediction [4,28,34]. However, they ignore the prediction bias and the workers' willingness. It will lead to the absence of selected workers, causing low assignment quality. Consider the following example: Referring to workers' historical trajectories, the SC platform finds that Bob is often near McDonald's at around 12:30 p.m. on working days, so the platform plans to assign Bob to the task, which is taking photos of the customer flow near the McDonald. But actually, Bob has applied for half a day off because of a bad cold and driven home at 12:00 p.m., so he could not accomplish the task, and the platform should reassign it.

The above example shows that in practical situations, due to the absence of workers, the prediction-based spatial-temporal task assignment needs to be adjusted appropriately to ensure the reliability. Gummidi et al. also pointed that breaking the rule of immutability in spatial-temporal crowdsourcing task assignment would help to reallocate tasks, and it advantages to improve the total matching degree and task acceptance [11]. However, the following challenges still exist during the task assignment adjustment process. First, when should the spatial-temporal crowdsourcing tasks be reassigned? The platform may find workers will not perform the task at different times, so it is difficult to find a fixed time to reallocate the task. If the reassignment occurs after the start time of spatial-temporal task, other platforms may book the potential workers. Secondly, how to reassign spatial-temporal crowdsourcing tasks? The original assignment scheme has existed, an important problem is how to adjust the original assignment scheme properly to ensure the normal execution of all tasks.

To deal with the above challenges, this article proposes a Spatial-temporal Task Reassignment Method (STR) based on prediction, which consists of four modules: monitoring module, judgement module, candidate worker and task selection module, and task reassignment module. For the first challenge, when the number of workers for a spatial-temporal task is less than the lower boundary of task requirement, the monitoring module and the judgement module will work together to trigger a task reassignment process. The candidate worker and task selection module and the task reassignment module are designed for the second challenge. When task reassignment is triggered, the two module search for candidate tasks and workers, and then reassign the spatial-temporal tasks to minimize the adjustment to the original scheme and maximize the total matching degree.

The main contributions are summarized as follows:

– Propose a reassignment method for spatial-temporal tasks based on prediction. This method breaks the rule of immutability in spatial-temporal

task assignment based on prediction to improve the reliability of assignment scheme.
- To maximize the total matching degree and minimize the adjustment of the original assignment scheme, we employ the multi-objective algorithm to reassign the spatial-temporal tasks.
- We evaluate the proposed spatial-temporal task reassignment method on three different real data sets to verify its effectiveness.

The remainder of the article is organized as follows. In Sect. 2, we review related work on task assignment in SC. In Sect. 3, we first give an overview about basic concepts used in the article, and then define the assignment problem. Section 4 introduces the process of STR, and then presents each module in detail. Section 5 verifies the proposed method through experiments. In Sect. 6, we conclude the article and outline the future work.

2 Related Work

According to the availability of task and worker information, spatial-temporal crowdsourcing task assignment can be divided into two categories: online task assignment and offline task assignment [2].

Online task assignment means the spatial-temporal task is pre-submitted to the platform, but the worker enters the platform dynamically. The platform does not know which workers will log in before the task starts, but needs to decide whether to allocate spatial-temporal tasks to the workers immediately according to availability information [3,6,31]. Liu et al. recruited workers to perceive the traffic flow within the task sensing period, and then coarsely selected the most suitable road section for deployment according to the perception data [16]. Cheng et al. considered combining different task assignment platforms to jointly complete multiple tasks online. As each platform focuses on different areas, they complete tasks as much as possible by assisting each other and sharing remuneration [6]. Liu et al. dynamically adjusted assignment schemes in the online spatial-temporal task assignment process by constantly monitoring the information of spatial-temporal tasks and workers obtained currently [13,14].

In offline task assignment, the platform owns worker information before the task starts, and based on the historical records or prediction information to allocate the tasks to workers [1,12,30]. Offline spatial-temporal crowdsourcing task assignment mainly includes three methods: greedy strategy, task assignment based on graph model and task assignment based on machine learning method. Greedy strategy is widely used in offline spatial-temporal task assignment. Zhao et al. defined a task acceptance probability, and then in order to obtain multiple worker groups, they employed greedy strategy to calculate the maximum task acceptance probability. Workers in one group cooperate to complete spatial-temporal tasks. She et al. proposed the Greedy-GEACC algorithm. Workers select the tasks they are interested in, and ensure that the tasks do not conflict with each other [32]. They pointed out that the Greedy-GEACC algorithm has the advantage of strong scalability in large datasets [17]. Wei

et al. proposed a greedy GP-BS method to select the most effective matching pair of sensing region and workers by jointly considering the relationship among perceptual regions and worker mobility [25]. In the task assignment based on graph model, the spatial-temporal task assignment is transformed into minimum cost maximum flow (MCMF) problem. HCDT model was proposed in 2019 [33]. After obtaining the time preference tensor of workers, HCDT model converts the spatial-temporal tasks assignment into MCMF problem to obtain the optimal solution satisfying the execution distance constraints of workers and the time constraints of spatial-temporal tasks [35]. Machine learning is a hot research topic currently. The spatial-temporal task assignment also employs methods based on machine learning. Wang et al. proposed the EMOPSO algorithm, which uses an improved particle swarm optimization algorithm to obtain the assignment scheme to achieve the two objectives of task coverage maximization and platform cost minimization [24]. Zhu et al. proposed DLMV method to collect environmental information at designated points in the city [37]. They predicted vehicle trajectory by deep learning, and then assigned tasks to vehicles according to the predicted vehicle trajectory.

However, all the above research follows the rule of immutability during the task assignment by default, so they cannot deal with the failure of spatial-temporal task execution caused by the absence of workers. Some recent work has pointed the irrationality of the rule of immutability [7,11]. Cheng et al. proposed a series of greedy assignment methods to solve the problem that activities cannot be executed normally caused by the attribute information changes of workers or activities [7]. On the basis of satisfying the constraint of minimizing changes to the original allocation scheme as much as possible, the matching degree of all activities should be maximized.

Our work is different from the existing research in the following four aspects: 1) Compared with the online spatial-temporal task assignment method, the assignment in our work is before the task start time, which avoids potential worker losing due to insufficient workers. 2) Compared with the offline spatial-temporal task assignment method, our work focuses on the adjustment of the existing scheme, so it breaks the rule of immutability that most work follows. 3) Compared with the work of Cheng et al. [7], our reassignment method has two objectives, so it is more practical. 4) In the process of reassignment, we not only consider the effect of deleting edges from the original assignment graph, but also the effect of adding edges to the original assignment graph.

3 Problem Statement

In this section, we introduce a set of necessary preliminary concepts. Table 1 lists the definitions used in the article.

The worker set is W, $W = \{w_1, w_2, ..., w_{|W|}\}$, each worker has attribute set At_w. Specifically, At_w contains tag_w, ps_w, max_w and b_w. tag_w is the tag set preferred by the worker. ps_w shows the sequence of location and time tuples that workers have appeared, $ps_w = \{(\text{location}, \text{time}), ...\}$. max_w is the upper

Table 1. Symbols and descriptions.

Symbol	Description
s	A spatial-temporal task
At_s	Attribute set of task s
l_s	Location of task s
t_s^{st}	Start time of task s
t_s^{et}	End time of task s
tag_s	Tag set of task s
c_s	Cost of performing task s
ub_s	Upper boundary of workers required by s
lb_s	Lower boundary of workers required by s
(lo_s, la_s)	GPS coordinates of s location
S	Spatial-temporal task set
w	A worker
At_w	Attribute set of worker w
ps_w	Sequence of location and time tuples
tag_w	Tag set of task w
b_w	Budget of w
max_w	Upper boundary of tasks planned to complete by w
W	Worker set
γ	Trigger task
W_γ	Available worker set of γ
S_γ^c	Candidate task set of γ
W_γ^c	Candidate worker set of γ
$\mathbf{M}_{ij}$	Matching degree between w_i and s_j
$\mathbf{M}$	Matching degree Matrix
$\mathcal{G}$	Bipartite graph of original assignment scheme
$\mathcal{G}'$	Bipartite graph of reassignment scheme
$\mathcal{W}$	Worker node set in bipartite graph
$\mathcal{S}$	Task node set in bipartite graph
$\mathcal{E}$	Edge set in bipartite graph

boundary of the number of spatial-temporal tasks that the worker plans to complete. b_w is the worker budget. Similarly, the spatial-temporal task set is S. For each spatial-temporal crowdsourcing task s, At_s represents its attribute set, including task execution location (l_s), start time (t_s^{st}), end time (t_s^{et}), tag set (tag_s), upper boundary of workers required by the task (ub_s), lower boundary of workers required by the task (lb_s), and possible cost of completing the task (c_s). Locations are represented by GPS coordinates ((lo_s, la_s)).

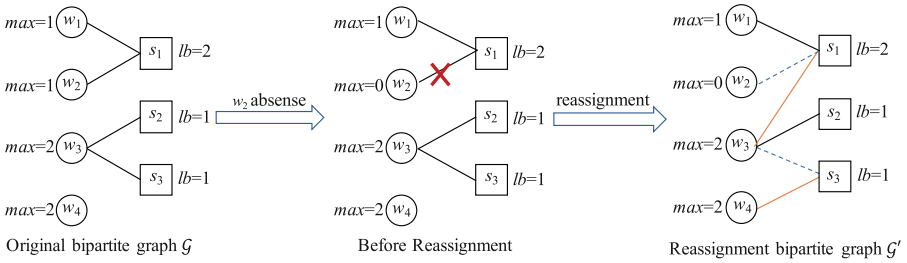


Fig. 1. An example of a bipartite graph of spatial-temporal task reassignment.

By mining and calculating worker trajectory and historical matching information, we predict the matching matrix of worker set and spatial-temporal task set, represented by $\mathbf{M}$, where each element $\mathbf{M}_{ij}$ is the matching degree of worker w_i and spatial-temporal task s_j. $\mathcal{G}$ represents the bipartite graph constituted by the original assignment scheme, and $\mathcal{G}'$ represents the bipartite graph after reassignment, as shown in Fig. 1. Due to the absence of w_2, the lower boundary of workers for s_1 is not met and needs to be reassigned. Therefore, s_1 becomes the trigger spatial-temporal task γ, W_γ is the set of available workers of γ, including all workers except w_2. Comparing with $\mathcal{G}$, $\mathcal{G}'$ deletes edges $e(w_2, s_1)$ and $e(w_3, s_3)$, adds edges $e(w_3, s_1)$ and $e(w_4, s_3)$. We employ $|\mathcal{G} - \mathcal{G}'|$ to represent the number of edges deleted from the original assignment graph, and $|\mathcal{G}' - \mathcal{G}|$ to represent the number of edges added to the original bipartite graph. Therefore, in Fig. 1, $|\mathcal{G} - \mathcal{G}'| = 2$, $|\mathcal{G}' - \mathcal{G}| = 2$.

Problem Formulation: Given the worker set W, spatial-temporal task set S, predicted workers and task matching matrix $\mathbf{M}$, and the current assignment scheme $\mathcal{G}$. When a task becomes a trigger task γ, we select the candidate worker set W_γ^c and candidate spatial-temporal task set S_γ^c from W and S respectively, and then reassign in the solution space composed of γ, W_γ^c and S_γ^c. The objectives of the reassignment scheme $\mathcal{G}'$ are to maximize the matching probability of all tasks and minimize the adjustment of the original allocation scheme under the constraints, as shown in Eq. (1)–Eq. (6):

$$maximize \quad \sum_{s_j \in S_\gamma^c \cup \gamma} \sum_{w_i \in W_\gamma^c} \mathbf{M}_{ij} \cdot \theta_{ij} \tag{1}$$

$$minimize \quad |\mathcal{G} - \mathcal{G}'| \times d + |\mathcal{G}' - \mathcal{G}| \times a \tag{2}$$

$$s.t. \quad \theta_{ij} = \begin{cases} 1, & \text{if} \quad e(w_i, s_j) \in \mathcal{G}'.\mathcal{E}, \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

$$\forall w_i \in W_\gamma^c, \quad \sum_{j=1}^{|S|} \theta_{ij} \leq max_{w_i}, \tag{4}$$

$$\forall s_j \in S_\gamma^c, \quad lb_{s_j} \leq \sum_{i=1}^{|W|} \theta_{ij} \leq ub_{s_j}. \tag{5}$$

$$\forall w_i \in W_\gamma^c, \quad \sum_{j=1}^{|S|} \theta_{ij} c_{s_j} \leq b_{w_i}, \tag{6}$$

where θ_{ij} indicates whether w_i has been assigned to s_j in $\mathcal{G}'$. $\theta_{ij} = 1$ means s_j will be allocated to w_i and vice versa (Eq. (3)). Adjustments to the original assignment scheme include deleting edges from the original assignment bipartite graph and adding edges to the graph. d in Eq. (2) represents the cost of deleting edges, and a means the cost of adding edges. Therefore, Eq. (2) reflects the objective of reducing the adjustment to the original assignment scheme as much as possible. Equation (4) and Eq. (5) show the limitation of the number of tasks that workers plan to participate in and the worker requirement limitation of spatial-temporal tasks, respectively. Equation (6) shows the budget constraint of workers.

4 Spatial-Temporal Task Reassignment Method

In this section, we first introduce the workflow of the reassignment method for spatial-temporal tasks, as shown in Fig. 2, to answer the two questions of when and how to do the reassignment. Then, we introduce the candidate worker and task selection module and the task reassignment module in detail.

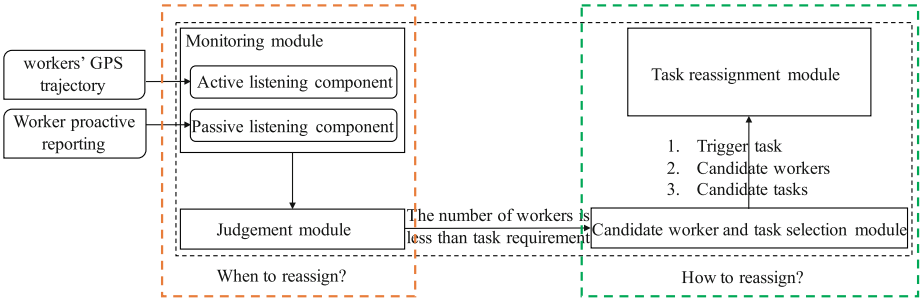


Fig. 2. STR method workflow.

4.1 Workflow of Spatial-Temporal Task Reassignment

As shown in Fig. 2, the platform uses the monitoring module to interact with the selected workers after formulating the original assignment scheme for spatial-temporal tasks. If the trajectories of workers can be monitored, the platform will collect the real-time GPS trajectory through the active listening component in

the monitoring module. When the platform finds that the reserved worker cannot reach the task location within the task sensing period, it transmits the worker ID to the judgement module. If it is difficult for the platform to collect the trajectories of workers, a passive listening component is used to collect workers' report, and then transmit the ID of the reserved worker to the judgement module. We believe the monitoring method is feasible, because if workers agree with the platform to collect trajectories, the platform does not need to disturb workers' normal activities to calculate whether workers can reach the task location within the task sensing period, and if it is difficult for the platform to collect worker trajectories, absent workers can actively report in advance. By interacting with the workers, the monitoring module finds the absent workers as early as possible to get sufficient time for the subsequent reassignment.

After sending the ID of absent workers and the ID of spatial-temporal tasks that may become trigger tasks to the judgement module, the judgement module measures whether the employers of the spatial-temporal task is less than the lower boundary of workers required by it. If it is less than the lower boundary, the spatial-temporal task is regarded as a trigger task. The absent worker IDs are transmitted to the candidate worker and task selection module to touch off the reassignment process. To reserve workers with high matching degree as much as possible and reduce the risk of being recruited by other spatial-temporal crowdsourcing platforms, tasks will be reassigned once the platform deems it as a trigger task.

In order to improve the efficiency of spatial-temporal task reassignment, we design the candidate worker and task selection module to select candidate task set S_γ^c and candidate worker set W_γ^c, and regard these two sets and the trigger task γ as the solution space to reduce the search space of reassignment scheme. After selecting the candidate workers and the candidate spatial-temporal tasks, the task reassignment module starts to work. The objectives of the reassignment scheme $\mathcal{G}'$ is to maximize the total matching degree and minimize the modification of the original allocation scheme under constraints.

4.2 Candidate Worker and Task Selection Module

In the candidate worker and task selection module, we use two methods to find suitable candidate worker set and spatial-temporal task set: matching degree sorting method and selecting similar workers (spatial-temporal task).

Matching Degree Sorting. Since one objective of reassignment is to maximize the total matching degree, it expects to select some candidate workers and candidate spatial-temporal tasks to ensure that the matching degree of candidate workers and γ is higher than with the original assignment task, i.e., candidate task. The matching degree sorting selection process is shown in Algorithm 1.

First, the candidate task set $S_\gamma^{c'}$ and the candidate worker set $W_\gamma^{c'}$ are initialized (line 1). Next, we initialize a list H to store pairs of workers and tasks in line 2. Then, for each spatial-temporal task s_j, check whether the number of reserved

Algorithm 1. Matching degree sorting

Input: **M**: matching degree matrix; S: spatial-temporal task set; γ: trigger task; $\mathcal{G}$: original assignment scheme; β: selecting number.
Output: $S_\gamma^{c'}$: candidate task set of γ; $W_\gamma^{c'}$: candidate worker set of γ.

1: Initialize $S_\gamma^{c'}$ and $W_\gamma^{c'}$: $S_\gamma^{c'} \leftarrow \varnothing$, $W_\gamma^{c'} \leftarrow \varnothing$;
2: Initialize a list H to store pairs of workers and tasks;
3: **for** $\forall s_j \in S$ **do**:
4: **if** $|e(w_*, s_j) \in \mathcal{G}.\mathcal{E}| > lb_{s_j}$ **then**:
5: **for** each w_i assigned to s_j in $\mathcal{G}$ **do**:
6: **if** w_i is not absent **then**:
7: Calculate $\Delta = \mathbf{M}_{i\gamma} - \mathbf{M}_{ij}$;
8: Insert w_i and s_j into H by descending order of Δ;
9: **end if**
10: **end for**
11: **end if**
12: **end for**
13: $S_\gamma^{c'} \leftarrow$ top β tasks in H;
14: $W_\gamma^{c'} \leftarrow$ top β workers in H;
15: **return** $S_\gamma^{c'}$ and $W_\gamma^{c'}$;

workers in the current assignment scheme is greater than the lower boundary of workers required by s_j. If so, it means that s_j can provide workers for the trigger task γ (lines 3 and 4), where w_* represents any worker node connected to s_j in $\mathcal{G}$. If the number of workers reserved for s_j is greater than the lower boundary of workers required by s_j, we calculate the matching degree gain Δ brought by reassigning the workers w_i to the trigger task γ (line 7). Then, insert (w_i, s_j) into the temporary storage list H according to the descending order of the matching degree gain (line 8). Finally, we select the first β spatial-temporal tasks from H as the candidate task set obtained by sorting method (line 13), and the first β workers as the candidate worker set obtained by sorting method (line 14).

Select Similar Candidate Workers and Spatial-Temporal Tasks. We select similar candidate workers from three perspectives: workers with similar spatial-temporal relation to the trigger task, workers with similar attributes to the absent worker, and workers of other tasks that have similar attributes to the trigger task. We will introduce the three approaches in detail. Algorithm 2 shows the process of selecting candidate workers and spatial-temporal tasks by similarity.

A) Workers who are similar to the trigger task. Because the spatial-temporal task has strict restrictions on sensing period and location, only the selected workers arrive at the designated location within the sensing period, task could be completed. Therefore, it is necessary to select workers who has similar spatial-temporal relationship to the trigger task. We mine the historical

Algorithm 2. Similarity ranking

Input: $\mathbf{M}$; W_γ; S; γ; $\mathcal{G}$; α.
Output: $S_\gamma^{c''}$: similar candidate task set; $W_\gamma^{c''}$: similar candidate worker set.

1: Initialize $S_\gamma^{c''}$ and $W_\gamma^{c''}$: $S_\gamma^{c''} \leftarrow \varnothing$, $W_\gamma^{c''} \leftarrow \varnothing$;
2: Initialize three storage list: H_1, H_2, H_3;
3: **for** $\forall w_i \in W_\gamma$ **do**:
4: Calculate the similarity between w_i and γ, and then store it into H_1;
5: Calculate the similarity between w_i and the absent worker, and store it intoH_2;
6: **end for**
7: Rank the workers in H_1 and H_2 in descending order, and select the first α workers from H_1, and then store them into $W_\gamma^{c''}$. Select the first α workers from H_2, who does not in $W_\gamma^{c''}$, and then store them into $W_\gamma^{c''}$;
8: **for** $\forall s_j \in S$ **do**:
9: Calculate the similarity, $sim_{s_j,\gamma}$, between s_j and γ, and store it into H_3;
10: **end for**
11: Sort tasks in H_3 in descending order;
12: **while** $\alpha > 0$ **do**:
13: $s_j = \mathrm{Pop}(H_3)$;
14: $W_\gamma^{c''} \leftarrow \{w_i | w_i \in \{e(w_*, s_j) \in \mathcal{G}.\mathcal{E}\} \quad \text{and} \quad w_i \notin W_\gamma^{c''}\}$;
15: $\alpha = \alpha - |\{w_i | w_i \in \{e(w_*, s_j) \in \mathcal{G}.\mathcal{E}\} \quad \text{and} \quad w_i \notin W_\gamma^{c''}\}|$;
16: **end while**
17: **for** $w_i \in W_\gamma^{c''}$ **do**:
18: Initialize storage list H;
19: **for** each s_j reserve w_i in $\mathcal{G}$ **do**:
20: Calculate $\Delta = \mathbf{M_{i\gamma}} - \mathbf{M_{ij}}$, and insert w_i and s_j into H by descending order of Δ;
21: **end for**
22: $S_\gamma^{c''} \leftarrow S_\gamma^{c''} \cup Pop(H)$;
23: **end for**
24: **return** $S_\gamma^{c''}$ and $W_\gamma^{c''}$;

trajectories and calculate the frequency of workers passing through the specified location during the trigger task sensing period. Then, select the first α available workers with high frequency to join the candidate worker set $W_\gamma^{c''}$. The process is shown in Lines 4 and 7 of Algorithm 2.

B) Workers with similar attributes to the absent workers. Workers' historical trajectories, preferences and execution records play a decisive role in predicting whether they can complete the spatial-temporal task. If the workers could not reach the specified location within the sensing period, they would not be able to complete the spatial-temporal task. Similarly, if the workers are not interested in the task, they would not execute it. Therefore, selecting workers with similar attributes to the absent workers can improve the suitability of candidate workers in three aspects: time, space and preference, and this method effectively reduces the search space. Specifically, we get the spatial-temporal matrix $\mathbf{LT}_{w_{i'}}$ of all absent workers of the triggering task γ

and the spatial-temporal matrix $\mathbf{LT}_{w_i}$ of available worker w_i by prediction [20,26], and then design Eq. (7) to calculate the spatial-temporal similarity between the absent workers and an available worker w_i, which is shown by $sim^{lt}_{w_i}$:

$$sim^{lt}_{w_i} = \left\| (\sum_{w_{i'} \in W_{ab}} \mathbf{LT}_{w_{i'}}) - \mathbf{LT}_{w_i} \right\|_2, \tag{7}$$

The W_{ab} is the absent worker set of the trigger task γ. Then, we calculate the similarity between the tag set of absent workers and w_i by set similarity, as shown in Eq. (8):

$$sim^{tag}_{w_i} = \frac{(\bigcup_{w_{i'} \in W_{ab}} tag_{w_{i'}}) \cap tag_{w_i}}{(\bigcup_{w_{i'} \in W_{ab}} tag_{w_{i'}}) \cup tag_{w_i}}, \tag{8}$$

Therefore, the similarity between the absent workers and w_i is:

$$sim_{w_i} = sim^{lt}_{w_i} + sim^{tag}_{w_i}, \tag{9}$$

After obtaining the similarity of available workers and the absent workers, we select the first α workers with the high similarity and add them into the similar candidate worker set $W^{c''}_{\gamma}$. As shown in lines 5 and 7 of Algorithm 2.

C) Workers of other tasks that have similar attributes to trigger tasks. If the attributes of a task s_j are similar to the trigger task γ, the workers performing s_j may have a high matching degree with γ. In order to find other similar tasks with γ, we measure the attribute similarity from four dimensions: location, time, tag and other scalar attributes. Intuitively, if the distance between s_j and γ is closer, the location similarity between them is higher. Therefore, we design Eq. (10) as the attenuation function to represent the similarity, and it will decrease as distance increases.

$$sim^{dist}_{s_j,\gamma} = e^{-g \cdot dist(s_j,\gamma)}, \tag{10}$$

where $dist(s_j, \gamma)$ is the distance function used to calculate the distance between γ and s_j, and g is a hyperparameter. $sim^{dist}_{s_j,\gamma}$ belongs to the interval of $(0, 1]$. For the time attribute of two spatial-temporal tasks, the more overlapping sensing period, the more similar they will be. Therefore, we employ the set similarity method to calculate the time similarity between γ and s_j.

$$sim^{time}_{s_j,\gamma} = \frac{\{t^{st}_{s_j}, t^{st}_{s_j} + 1, \cdots, t^{et}_{s_j}\} \cap \{t^{st}_{\gamma}, \cdots, t^{et}_{\gamma}\}}{\{t^{st}_{s_j}, t^{st}_{s_j} + 1, \cdots, t^{et}_{s_j}\} \cup \{t^{st}_{\gamma}, \cdots, t^{et}_{\gamma}\}}, \tag{11}$$

If the two tasks have more identical tags, they will be more similar in tags. The calculation formula of tag similarity is as follows:

$$sim^{tag}_{s_j,\gamma} = \frac{tag_{s_j} \cap tag_{\gamma}}{tag_{s_j} \cup tag_{\gamma}}, \tag{12}$$

Since other attributes of spatial-temporal tasks are scalars, we use a tuple to contain them. We calculate the similarity between attribute tuples of different tasks by L2 normal form.

$$sim_{s_j,\gamma}^{other} = \left\| (ub_{s_j}, lb_{s_j}, c_{s_j}) - (ub_\gamma, lb_\gamma, c_\gamma) \right\|_2, \tag{13}$$

Therefore, the similarity between the attributes of γ and s_j can be obtained.

$$sim_{s_j,\gamma} = sim_{s_j,\gamma}^{dist} + sim_{s_j,\gamma}^{time} + sim_{s_j,\gamma}^{tag} + sim_{s_j,\gamma}^{other}. \tag{14}$$

We sort the spatial-temporal tasks according to attribute similarity in descending order, and the first α workers who are not in the candidate worker set are selected and added into the similar candidate worker set $W_\gamma^{c''}$. As shown in lines 8 to 16 of Algorithm 2.

The same as matching degree sorting, to get similar task candidate set, we check the original assignment scheme of each candidate worker,i.e., w_i. If w_i performing γ brings a greater gain in matching degree than performing the originally assigned task s_j, we add s_j into the temporary storage list H. The spatial-temporal tasks in H are sorted according to the gain of matching degree. The s_j that brings the maximum matching degree gain is taken from H and added into the candidate spatial-temporal task set $S_\gamma^{c''}$. As shown in lines 17 through 23 of Algorithm 2.

Candidate Workers and Candidate Spatial-Temporal Tasks. The final set of candidate workers is $W_\gamma^c = W_\gamma^{c'} \cup W_\gamma^{c''}$. Similarly, we obtain the final set of candidate spatial-temporal tasks by $S_\gamma^c = S_\gamma^{c'} \cup S_\gamma^{c''}$.

4.3 Task Reassignment Module

As the problem definition in Eq. (1)–Eq. (6), there are two objectives during reassignment, which are maximizing the matching degree of all tasks and minimizing the changes to the original assignment scheme, so we seek multi-objective optimization methods to achieve the reassignment. In addition, since the search space of the reassignment has been greatly reduced by the candidate worker and task selection module, only including γ, the candidate worker set W_γ^c, and the spatial-temporal task set S_γ^c. In order to obtain the global optimal assignment scheme as far as possible, we explore several swarm intelligence algorithms which can solve the multi-objective problem in the task reassignment module.

We try to integrated many multi-objective swarm intelligence algorithms, such as awGA [10], NSGA2 [9], MOEAD [29], NSGA3 [8] and RVEA [5], and decide to employ NSGA3 as the default multi-objective reassignment method through experiments. Each individual on the Pareto front obtained by the NSGA3 is an optimal reassignment scheme, and we could select according to the actual demand.

5 Performance Evaluation

This section will answer the following questions through experiments: (1) Comparing the multi-objective swarm intelligence algorithms integrated into the task reassignment module, which algorithm is more suitable? (2) Comparing with the baselines, how much doe effectiveness of single trigger task reassignment improve by STR method? How much effectiveness does the two selecting candidate workers and tasks methods contribute, respectively? (3) How much influence of the adding edges cost and the deleting edges cost on the original bipartite graph during reassignment? (4) Compared with the baselines, how much does effectiveness of dynamic multi-task reassignment improve by using STR method?

5.1 Dataset and Experimental Settings

We conduct extensive experiments on three real-world datasets, Geolife [36], F-NYC [27] and Meetup-HT [15]. Geolife is a trajectory intensive dataset. F-NYC is a check-in dataset, which includes sparse check-in records. Meetup-HT is an event-based social network dataset, which includes historical participation records. We follow the preprocessing method in [20,26] to get datasets and matching degree matrix obtained through prediction. The information of the three datasets is shown in Table 2, where $|\boldsymbol{R}|$ is the record number of all workers, and $|\boldsymbol{L}|$ is the number of locations in the dataset.

Table 2. Basic statistics of datasets.

Dataset	$\mid W \mid$	$\mid S \mid$	$\mid tag \mid$	$\mid R \mid$	$\mid L \mid$
Geolife	85	150	68	11091	114
F-NYC	272	600	6445	12	33150
Meetup-HT	644	1421	2981	9687	–

We choose the total matching degree (ToMD) [7,18] and adjustment cost (Cost) as the metrics of reassignment methods. ToMD shows the reliability of the reassignment scheme compared with the original scheme. According to the second objective function, the reassignment scheme should minimize adjustments to the original assignment strategy as much as possible, so we set Cost to monitor the adjustment. These two metrics provide a clear description about reassignment schemes on different objectives.

We integrate Geatpy[1], which is written by Python, into the spatial-temporal task reassignment module to achieve the multi-objective swarm intelligence algorithms, such as awGA, NSGA2, MOEAD, NSGA3 and RVEA. ξ-In [7], which is the most advanced spatial-temporal crowdsourcing task reassignment method, is adopted as a baseline. In order to reflect the necessity of the sorting method

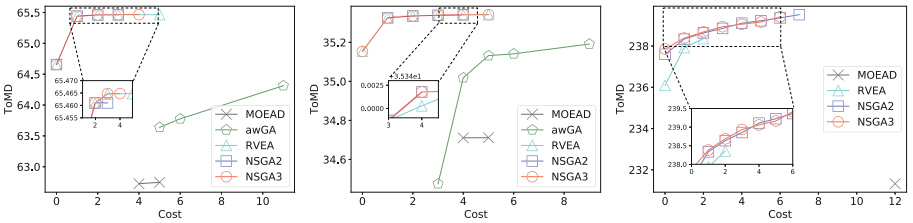
[1] http://geatpy.com/.

and the similarity method during candidate worker and spatial-temporal task selection process, we design two additional comparison methods: STR-Sort and STR-Sim, which only use sorting method or similarity method to select candidate workers and tasks, respectively. In addition, in order to reflect the difference between the original allocation scheme and the reassignment scheme facing absent workers, we use Non-reass to represent the total matching degree after worker absence but without reassignment. As the ξ-In algorithm only considers the effect of removing edges from the original allocation scheme and deems it as a constraint, to compare with ξ-In, the default value of removing edges cost (d) is 1, and adding edges cost (a) is 0.

All experiments are performed on the same hardware environment, equipped with Intel (R) Xeon (R) CPU E5-2680 v4@2.40GHz and 128 GB memory.

5.2 Comparison of Multi-objective Reassignment Algorithms

In order to answer the first question, we compare the swarm intelligence algorithms mentioned in Sect. 4.3 in the spatial-temporal crowdsourcing task reassignment. Figure 3 shows the Pareto fronts of different multi-objective algorithms.



(a) Pareto frontiers of different multi-objective optimization algorithms on Geolife

(b) Pareto frontiers of different multi-objective optimization algorithms on F-NYC

(c) Pareto frontiers of different multi-objective optimization algorithms on Meetup-HT

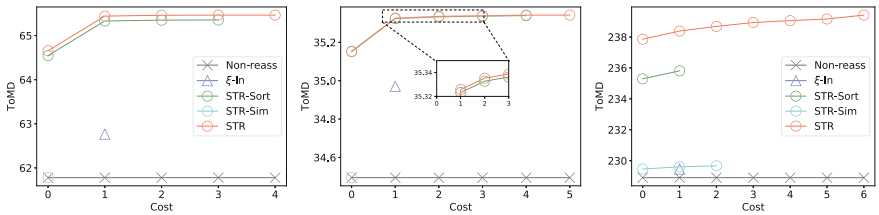
Fig. 3. Comparison of different multi-objective algorithms in the task reassignment module.

In Fig. 3, both NSGA2 and NSGA3 algorithms present ascendant Pareto fronts in the three datasets. We find that NSGA3 algorithm is slightly better than NSGA2 in the local magnification, because NSGA3 preserves more population diversity in the process of selecting the next generation. Therefore, we select NSGA3 as the default reassignment algorithm in the spatial-temporal task reassignment module. The Pareto front of RVEA algorithm is slightly lower than NSGA2 and NSGA3, which may be because RVEA algorithm is suitable for high-dimensional space composed of more objective functions. Compared with the above three algorithms, awGA and MOEAD algorithms do not find the solution when the Cost is 0 or 1, and in the case of overspending, the total matching degree is lower than NSGA3, NSGA2 and RVEA algorithms. In addition, MOEAD algorithm fails to find a feasible allocation scheme on the

Meetup-HT dataset. Therefore, in the task reassignment module, the algorithm that uses non-dominated sorting to solve the multi-objective problem is more suitable than the algorithm that transforms the multiple objectives into a single objective.

5.3 Comparison of Single Trigger Task Reassignment

To answer the second question, we randomly select reserved workers as absent workers. When a spatial-temporal task becomes a trigger task, the platform will reassign it. We compare the STR method with all of the baselines, and show the results in Fig. 4.



(a) The impact of different reassignment methods on the two evaluation metrics on Geolife

(b) The impact of different reassignment methods on the two evaluation metrics on F-NYC

(c) The impact of different reassignment methods on the two evaluation metrics on Meetup-HT

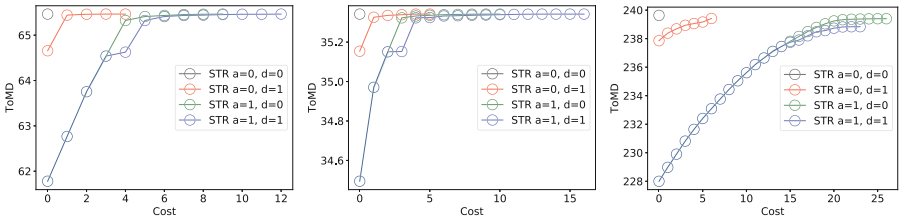
Fig. 4. Comparison of single trigger task reassignment by different methods.

The ToMD of Non-reass method is the lowest among all the methods on the three datasets, because the trigger task γ cannot be completed due to the absence of workers. According to the original assignment scheme, we calculate the total matching degree by spatial-temporal tasks except γ. Since the ξ-In reassignment method has one objective, we obtain only one reallocation scheme, as shown in Fig. 4, where is the blue triangle. Comparing with Non-reass method, the ToMD improves dramatically, but ξ-In deletes edges on the original allocation scheme, so the Cost is not 0 on all three datasets, and its ToMD is lower than STR method. Then discuss STR-Sort and STR-Sim methods. Comparing with STR-Sim, both of the STR-Sort and STR bring higher ToMD under the same Cost, but the Pareto front of STR-Sort is lightly lower than STR on the three data sets. Therefore, we consider that STR-Sort contributes more to STR, and STR-Sim could provides supplement to STR-Sort on selecting candidate workers and spatial-temporal task.

5.4 The Cost of Different Operational on the Reassignment Scheme

To answer the third question, we set four cost coefficient for evaluating the second objective, which are $a = 0, d = 0$; $a = 0, d = 1$; $a = 1, d = 0$ and $a = 1, d = 1$. $a = 0, d = 0$ means that we do not consider the cost of adjustment to the original scheme, which indicates the reassignment process does not consider the

second objective. $a = 0, d = 1$ only focuses on the cost of deleting edges from the original assignment bipartite graph, which is consistent with the ξ-In method. $a = 1, d = 0$ indicates only concern about the cost of adding edges to the original assignment bipartite graph. $a = 1, d = 1$ means that the cost includes adding and deleting edges on the original assignment graph.



(a) Impact of the cost of different operations on Geolife

(b) Impact of the cost of different operations on F-NYC

(c) Impact of the cost of different operations on Meetup-HT

Fig. 5. The influence of the cost of different operations during task reassignment.

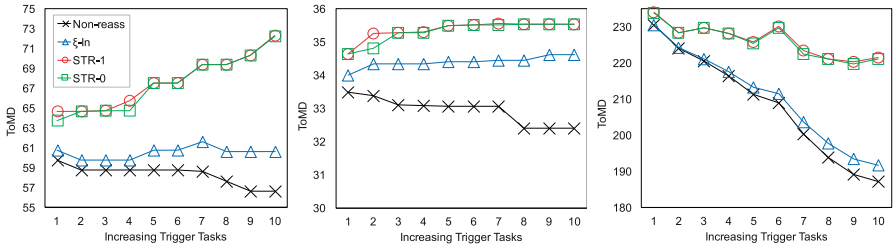
As shown in Fig. 5, $a = 0, d = 0$ is equivalent to only considering the first objective during the reassignment process, which does not concern the cost from changing assignment scheme, so it represents the highest ToMD. When set the cost coefficient according to the other three cases, the total matching degree of the STR method improves gradually as the total Cost increases. Among them, the Pareto front obtained by setting $a = 0, d = 1$ is more advanced than that obtained by setting $a = 1, d = 0$. It proves the effectiveness of the original assignment scheme, i.e., the platform assign spatial-temporal tasks to the workers with high matching degree. Therefore, the STR method is more willing to assign trigger task to the workers with residual execution capacity rather than snatch workers from other tasks during reassignment. When the cost coefficient is $a = 1, d = 1$, since both the deletion and addition of edges to the original scheme cause an increasing Cost, the Pareto front obtained by STR method with this setting is the lowest among the four coefficient settings.

5.5 Comparison of Dynamic Multi-task Reassignment

Next, we explore the case of reassigning multiple trigger tasks in succession. Due to space limitations, we show when the Cost is 0 and 1, the curve of the total matching degree changing with the increasing number of trigger tasks, and define them as STR-0 and STR-1, respectively. The results are represented in Fig. 6.

The Non-reass curve shows if the platform does not reassign the trigger task, the total matching degree will gradually decrease, so it is necessary to reassign trigger tasks. Comparing the ToMD curve of ξ-In and Non-reass, we find that the reassignment method is effective as the number of trigger tasks increases. Then, in Fig. 6, both of the total matching degree curves of STR-0 and STR-1 are higher

than that of ξ-In method on the three datasets, which shows the effectiveness of our STR. In addition, we find that the curve of STR-1 is slightly higher than STR-0. It proves swap scheme improves the total matching degree during the reassignment process, where the swap scheme means removing a certain worker from one spatial-temporal task, and assigning this worker to the trigger task.



(a) Reassignment of continuous multi trigger tasks on Geolife

(b) Reassignment of continuous multi trigger tasks on F-NYC

(c) Reassignment of continuous multi trigger tasks on Meetup-HT

Fig. 6. Comparison of Dynamic Multi-task reassignment by different methods.

6 Conclusion

In this article, to adjust task allocation scheme dynamically, we proposed a task reassignment method based on prediction in spatial-temporal crowdsourcing, which names STR. In STR, monitoring module and judgement module were used to determine whether trigger a task reassignment process. Once a task become a trigger task, we need select candidate workers and tasks by sorting and similarity methods to reduce the search space. Then, we formulated the reassignment as a multi-objective optimization problem, and employed swarm intelligence algorithms to solve it. Sufficient experiments built on three real datasets, and the results showed that comparing with the baselines, the STR method improves the total matching degree by about 10% while minimizing the adjustment on the original assignment scheme.

Although the proposed reassignment method can improve the total matching degree by about 10%, several further studies to make it more powerful are worth mentioning. How to balance the decision time and assignment quality will be considered in future work. Swarm intelligence algorithm has global optimal guarantee, but it often has a slow rate of convergence when dealing with a large number of participants and tasks. On the contrary, greedy strategies have advantages of short execution time when dealing with large-scale tasks, but can only obtain local optimal solutions. Therefore, how to balance assignment quality and decision time is worth exploring in the future.

Acknowledgements. This research was supported by the National Natural Science Foundation of China (62206105), the Natural Science Foundation of Jilin Province (20210101172JC, 20210101181JC), and the Fundamental Research Funds for the Central Universities, JLU.

References

1. Bei, X., Zhang, S.: Algorithms for trip-vehicle assignment in ride-sharing. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, pp. 3–9 (2018)
2. Chen, Y., Lv, P., Guo, D., Zhou, T.: A survey on task and participant matching in mobile crowd sensing. J. Comput. Sci. Technol. **33**(4), 768–791 (2018)
3. Chen, Z., Cheng, P., Zeng, Y., Chen, L.: Minimizing maximum delay of task assignment in spatial crowdsourcing. In: Proceedings of the 35th IEEE International Conference on Data Engineering, pp. 1454–1465 (2019)
4. Cheng, P., Lian, X., Chen, L., Shahabi, C.: Prediction-based task assignment in spatial crowdsourcing. In: Proceedings of the 33rd IEEE International Conference on Data Engineering, pp. 997–1008 (2017)
5. Cheng, R., Jin, Y., Olhofer, M., et al.: A reference vector guided evolutionary algorithm for many-objective optimization. IEEE Trans. Evol. Comput. **20**(5), 773–791 (2016)
6. Cheng, Y., Li, B., Zhou, X., Yuan, Y., Wang, G.: Real-time cross online matching in spatial crowdsourcing. In: Proceedings of the 36th IEEE International Conference on Data Engineering, pp. 1–12 (2020)
7. Cheng, Y., Yuan, Y., Chen, L., Giraud-Carrier, C.: Event-participant and incremental planning over event-based social networks. IEEE/ACM Trans. Knowl. Data Eng. **33**(2), 474–488 (2021)
8. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: solving problems with box constraints. IEEE Trans. Evol. Comput. **18**(4), 577–601 (2014)
9. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
10. Gen, M., Cheng, R.: Genetic Algorithms and Engineering Optimization. Wiley, New York (2000)
11. Gummidi, S.R.B., Xie, X., Pedersen, T.B.: A survey of spatial crowdsourcing. ACM Trans. Database Syst. **44**(2) (2019)
12. Liu, C.H., Piao, C., Ma, X., Yuan, Y.: Modeling citywide crowd flows using attentive convolutional LSTM. In: Proceedings of the 37th IEEE International Conference on Data Engineering, pp. 217–228 (2021)
13. Liu, W., Yang, Y., Wang, E., Wang, H.: Dynamic online user recruitment with (non-) submodular utility in mobile crowdsensing. IEEE/ACM Trans. Netw. **29**(5), 2156–2169 (2021)
14. Liu, W., Yang, Y., Wang, E., Wu, J.: Dynamic user recruitment with truthful pricing for mobile crowdsensing. In: 39th IEEE IEEE Conference on Computer Communications, pp. 1113–1122 (2020)
15. Liu, X., He, Q., Tian, Y., Lee, W., McPherson, J., Han, J.: Event-based social networks: linking the online and offline social worlds. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1032–1040 (2012)
16. Liu, Z., Chen, L., Tong, Y.: Realtime traffic speed estimation with sparse crowdsourced data. In: Proceedings of the 34th IEEE International Conference on Data Engineering, pp. 329–340 (2018)
17. She, J., Tong, T., Chen, L.: Conflict-aware event-participant arrangement. In: Proceedings of the 31st IEEE International Conference on Data Engineering, pp. 735–746 (2015)

18. She, J., Tong, Y., Chen, L.: Utility-aware social event-participant planning. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, pp. 1629–1643 (2015)
19. Song, T., Tong, Y., Wang, L., She, J.: Trichromatic online matching in real-time spatial crowdsourcing. In: Proceedings of the 33rd IEEE International Conference on Data Engineering, pp. 1009–1020 (2017)
20. Sun, B., Wei, X., Cui, J., et al.: Social activity matching with graph neural network in event-based social networks. Int. J. Mach. Learn. Cybern. **14**(6), 1989–2005 (2023)
21. Tong, Y., She, J., Ding, B., Wang, L., Chen, L.: Online mobile micro-task allocation in spatial crowdsourcing. In: Proceedings of the 32nd IEEE International Conference on Data Engineering, pp. 49–60 (2016)
22. Tong, Y., Wang, L., Zhou, Z., Ding, B.: Flexible online task assignment in real-time spatial data. Proc. VLDB Endow. **10**(11), 1334–1345 (2017)
23. Tong, Y., Zeng, Y., Ding, B., Wang, L., Chen, L.: Two-sided online micro-task assignment in spatial crowdsourcing. IEEE Trans. Knowl. Data Eng. **5**(2), 2295–2309 (2021)
24. Wang, L., Yu, Z., Han, Q., Guo, B.: Multi-objective optimization based allocation of heterogeneous spatial crowdsourcing tasks. IEEE Trans. Mob. Comput. **17**(7), 1637–1650 (2018)
25. Wei, X., Li, Z., Liu, Y., Gao, S.: SDLSC-TA: subarea division learning based task allocation in sparse mobile crowdsensing. IEEE Trans. Emerg. Top. Comput. **9**(3), 1344–1358 (2021)
26. Wei, X., Sun, B., Cui, J., et al.: Location-and-preference joint prediction for task assignment in spatial crowdsourcing. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **42**(3), 928–941 (2023)
27. Yang, D., Zhang, D., Zheng, V.W., Yu, Z.: Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. IEEE Trans. Syst. Man Cybern. Syst. **45**(1), 129–142 (2015)
28. Yang, Y., Liu, W., Wang, E., Wu, J.: A prediction-based user selection framework for heterogeneous mobile crowdsensing. IEEE Trans. Mob. Comput. **18**(11), 2460–2473 (2019)
29. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **11**(6), 712–731 (2007)
30. Zhang, X., Yang, Z., Liu, Y., Tang, S.: On reliable task assignment for spatial crowdsourcing. IEEE Trans. Emerg. Top. Comput. **7**(1), 174–186 (2019)
31. Zhao, B., Xu, P., Shi, Y., Tong, Y.: Preference-aware task assignment in on-demand taxi dispatching: an online stable matching approach. In: Proceedings of the 33rd AAAI Conference on Artificial Intelligence, pp. 2245–2252 (2019)
32. Zhao, Y., Guo, J., Chen, X., Hao, J.: Coalition-based task assignment in spatial crowdsourcing. In: Proceedings of the 37th IEEE International Conference on Data Engineering, pp. 241–252 (2021)
33. Zhao, Y., Xia, J., Liu, G., Su, H.: Preference-aware task assignment in spatial crowdsourcing. In: Proceedings of the 33rd AAAI Conference on Artificial Intelligence, pp. 2629–2636 (2019)
34. Zhao, Y., Zheng, K., Cui, Y., Su, H.: Predictive task assignment in spatial crowdsourcing: a data-driven approach. In: 36th International Conference on Data Engineering, pp. 13–24 (2020)
35. Zhao, Y., Zheng, K., Yin, H., Liu, G.: Preference-aware task assignment in spatial crowdsourcing: from individuals to groups. IEEE Trans. Knowl. Data Eng. (2020)

36. Zheng, Y., Zhang, L., Xie, X., Ma, W.Y.: Mining interesting locations and travel sequences from GPS trajectories. In: Proceedings of the 18th International Conference on World Wide Web, pp. 79–800 (2009)
37. Zhu, X., Luo, Y., Liu, A., Tang, W.: A deep learning-based mobile crowdsensing scheme by predicting vehicle mobility. IEEE Trans. Intell. Transp. Syst. **22**(7), 4648–4659 (2021)