Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Spatial crowdsourcing task allocation for heterogeneous multi-task hybrid scenarios: a model-embedded role division approach*

Zhenhui FENG[1,2], Renbin XIAO‡[1], Mingzhi XIAO[3]

[1]*School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China*

[2]*Petro China Kunlun Gas Company Limited Hubei Branch, Wuhan 430077, China*

[3]*School of Information Science and Engineering, Wuchang Shouyi University, Wuhan 430064, China*

E-mail: feng_zh@hust.edu.cn; rbxiao@hust.edu.cn; 2024111014@wsyu.edu.cn

Received Jan. 15, 2025; Revision accepted May 5, 2025; Crosschecked June 16, 2025; Published online July 5, 2025

Abstract: Spatial crowdsourcing (SC), as an effective paradigm for accomplishing spatiotemporal tasks, has gradually attracted widespread attention from both industry and academia. With the advancement of mobile technology, the service modes of SC have become more diversified and flexible, aiming to better meet the variable requirements of users. However, most research has focused on homogeneous task allocation problems under a single service model, without considering the individual differences among task requirements and workers. Consequently, many of these studies fail to achieve satisfactory outcomes in real scenarios. Based on real service scenarios, in this study, we investigate a heterogeneous multi-task allocation (HMTA) problem for hybrid scenarios and provide a formal description and definition of the problem. To solve the problem, we propose a role division approach embedded with an individual sorting model (RD-ISM). This approach is implemented based on a batch-based mode (BBM) and consists of two parts. First, an individual sorting model is introduced to determine the sequence of objects based on spatiotemporal attributes, prioritizing tasks and workers. Second, a role division model is designed based on an attraction–repulsion mechanism to match tasks and workers. Following several iterations over multiple batches, the approach obtains the final matching results. The effectiveness of the approach is verified using real and synthetic datasets and its performance is demonstrated through comparisons with other algorithms. Additionally, the impact of different parameters within the approach is investigated, confirming its scalability.

Key words: Spatial crowdsourcing (SC); Heterogeneous task; Role division; Attraction–repulsion mechanism; Individual sorting
https://doi.org/10.1631/FITEE.2500035 **CLC number:** TP399

1 Introduction

In recent years, with the development of information technology and the rapid popularization of mobile terminal devices, spatial crowdsourcing (SC) has become an effective model for accomplishing spatiotemporal tasks in the real world (Guo et al., 2018). This approach has gradually attracted widespread attention from both industry and academia (Wang L et al., 2018). SC leverages the mobile Internet to integrate and schedule idle offline resources through online recruitment, aiming for efficient sharing of resources (Chen et al., 2014). By using SC platforms, mobile workers can be recruited to handle various complex, large-scale, and distributed spatiotemporal tasks that are difficult for computers or a few experts to accomplish (Gong et al., 2020).

With the rapid development of the online-to-offline business models, SC has been widely applied across

‡ Corresponding author

* Project supported by the National Science and Technology Innovation 2030 Major Project of the Ministry of Science and Technology of China (No. 2018AAA0101200)

ORCID: Zhenhui FENG, https://orcid.org/0000-0001-7004-8868; Renbin XIAO, https://orcid.org/0000-0003-0951-2734; Mingzhi XIAO, https://orcid.org/0009-0000-2547-9626

various industries (Tong et al., 2017). These diverse SC platforms are gradually becoming integrated into people's daily lives, playing a significant role in areas like social governance and traffic management. In terms of transportation management, the time constraints and spatial location changes involved in residents' travel can be regarded as typical spatiotemporal tasks (Mazzetto, 2024). Shared mobility platforms of SC have reshaped the urban transportation industry ecosystem, and numerous related SC platforms have emerged, such as Uber, DiDi, and Gofun (Ray et al., 2023). According to the urban traffic operation report released by DiDi, the number of daily trips facilitated by the platform has exceeded 30 million people (Feng and Xiao, 2024).

As more users post tasks on crowdsourcing platforms, both the number and complexity of tasks are increasing, necessitating a large number of workers to complete these tasks. Task allocation is one of the key issues for SC platforms (Hettiachchi et al., 2022). The results of task allocation determine the service quality of the platform. An effective allocation mechanism can improve work efficiency, thereby increasing workers' income and saving platform costs. Task allocation in SC typically relies on a centralized platform that collects task object information and matches workers and tasks according to some assignment strategies (Alamri, 2024). Currently, most research has focused on offline task allocation problems, and numerous algorithms aimed at maximizing the utility and quantity of task allocation have been proposed (Zhang et al., 2023). In these studies, all task and worker data were assumed to be known at the outset algorithm, and these known data were used for offline (static environment) task allocation. However, in reality, workers and tasks emerge randomly, and platforms cannot obtain detailed information about tasks and workers in advance (Zhao BM et al., 2019). Therefore, offline algorithms do not perform well in actual application scenarios. Moreover, most previous research on crowdsourcing task allocation has focused on allocation problems under a single task type, where the tasks and worker attributes perceived by the crowdsourcing platform are completely identical (Wang YJ et al., 2019). However, in actual task allocation processes, tasks submitted by different users are often not exactly the same, and worker attributes also vary. Most SC allocation processes belong to heterogeneous multi-task allocation (HMTA) problems in hybrid scenarios.

Taking the online car-hailing scenario as an example, in the service process, users submit travel tasks to the platform online based on their individual requirements (You et al., 2023). The platform matches suitable workers (i.e., vehicles) to the users based on the task requirements and spatiotemporal information. Previous studies have generally treated travel tasks submitted by users as homogeneous and assumed that all vehicles have identical attributes (Tong et al., 2021; Cai et al., 2023). However, in reality, users' requirements are diverse, and vehicle types vary (Feng and Xiao, 2024). Currently, travel platforms such as DiDi offer users more personalized travel services and vehicle options, such as Express, Taxi, and Premier. Users can select different vehicle types based on their requirements, considering various factors, and even choose multiple options simultaneously. Therefore, the tasks submitted by users have personalized characteristics, and the platform needs to match an appropriate vehicle type according to the task requirements.

In summary, there are some shortcomings in the research on SC task allocation, mainly in the following aspects:

1. Most research was based on static scenarios, where the spatiotemporal information of all crowdsourcing tasks and workers is fully known before task allocation (Song et al., 2017b). However, in reality, there is unpredictability. The results of research based on static scenarios lack feasibility in practical applications. Therefore, it is necessary to study online task allocation in dynamic environments.

2. Most research has focused on homogeneous task allocation problems, where the tasks and worker attributes perceived by the crowdsourcing platform are assumed to be identical. However, in real task allocation processes, tasks submitted by different users are often not identical, and worker attributes vary. Therefore, the allocation process in most SC scenarios is characterized as an HMTA problem in a hybrid environment.

In response to the above challenges, in this study, we investigate a class of HMTA problems under the background of SC. Based on this, we design a quasi-real-time online task allocation method. Most existing methods target static allocation problems

(Zhou et al., 2023), while few approaches exist for solving dynamic task allocation problems (Tong et al., 2020). Additionally, in terms of task types, most existing methods consider single tasks, where the tasks and worker attributes are identical, while methods for solving heterogeneous task allocation problems are relatively scarce.

In the online task allocation problem, the sets of tasks and workers are initially unknown, and it is assumed that tasks and workers enter the platform in an uncertain order. Subsequently, the platform irrevocably assigns tasks to workers based on the principle of invariance (Tong et al., 2020); i.e., once a task allocation is implemented, it cannot be changed. Each allocation generates a corresponding revenue. The key to solving this problem is to design an allocation strategy that maximizes total profit while satisfying various practical constraints, such as spatial, time, and capacity constraints (Duan et al., 2019). In recent years, research on online task allocation has increased. For instance, Hassan and Curry (2014) studied the online task allocation problem, where users submitted demands in real time and the platform matched suitable candidates by analyzing the spatial information of workers. Zhao BM et al. (2019) proposed and investigated a stable online matching problem, which maximized revenue and user satisfaction by considering spatial distance and order value. Zhang et al. (2022) addressed the online allocation problem of multi-stage tasks in complex scenarios and developed a cross-regional task allocation algorithm based on offline data information. Song et al. (2017b) focused on workplaces as matching objects, studied online matching problems in specific environments, and proposed an immediate matching algorithm. Tong et al. (2021) considered the micro-task allocation problem GOMA in a new global online SC scenario and proposed a TGOA-Greedy algorithm based on a greedy strategy. Zhang et al. (2023) studied the optimal bilateral online allocation of tasks and workers under different arrival orders, solving the optimal bilateral matching problem and proposing an algorithm to maximize total revenue. In recent years, heterogeneous task allocation in SC has attracted growing attention from researchers. For example, Lin et al. (2024) investigated the dual-heterogeneous task allocation problem, proving that heterogeneous task allocation in SC is an NP-hard problem, and proposed a dual-heterogeneous task allocation algorithm to maximize task completion quality and minimize average travel distance. Bhatti et al. (2021) studied the dual-heterogeneous task allocation problem and proposed an approximate allocation scheme based on the partitioning and shifting method.

Due to the requirements of real-time processing, many methods commonly used for offline problems, such as exact algorithms and heuristic algorithms (Xiao et al., 2023), struggle to meet the efficiency demands of online problems. Although some online algorithms can address the above problems, constraints on spatiotemporal information and the limitations of task heterogeneity often lead algorithms to be trapped in local optima (Gong et al., 2020). Since online algorithms are highly sensitive to spatiotemporal information, it is necessary to conduct a reasonable analysis of this information in online algorithms to maximize the allocation utility and success rate (Liang et al., 2023). Most online task allocation algorithms have focused on optimizing allocation utility, neglecting the impact of time and space factors on the allocation results (Feng and Xiao, 2023), which results in a low success rate of task allocation. In summary, the main challenges in solving the problem studied in this paper are reflected in the following aspects:

1. In SC, the scale of the problem scenario is large (with urban areas as the unit), and it is an NP-hard problem when considering a dynamic environment. The platform requires allocation solutions within an acceptable time, making it significantly more challenging than typical task allocation problems.

2. Tasks and workers emerge randomly, and the matching process adheres to the principle of invariance. Each match affects subsequent task allocations. Therefore, it is necessary to weigh the current matching results at each step, leading to a coupled and complex solution process.

3. In hybrid scenarios, both tasks and workers exhibit individual differences, making this a heterogeneous task allocation problem with more constraints and greater difficulty.

To solve the above problems, this study is inspired by the role division phenomenon in group behavior of organisms (Xiao, 2024), and a quasi-real-time solution is designed by collective intelligence (CI). SC can be regarded as a practical application form of the

theory of CI (Wang MZ et al., 2022). Accordingly, the CI method can be used to further improve the task allocation efficiency of SC. In the field of CI, there are many methods for solving the task allocation problem, among which labor division and role division are typical. Labor division is inspired by lower organisms and is applicable to simple task allocation problems such as unmanned aerial vehicle (UAV) task allocation (Dortheimer, 2022), while role division is inspired by higher organisms and is a cooperative division mode based on group perception and group interaction, which can be used to solve task allocation problems in complex dynamic environments. During the role division process, individuals follow an attraction–repulsion mechanism to choose tasks. Different types of roles undertake different tasks, and each individual flexibly adjusts its role state based on attraction and repulsion, ultimately leading to an emergent overall role distribution (Wu et al., 2018). In this paper, we propose a role division approach embedded with an individual sorting model (RD-ISM) to solve the above problem. First, the objects to be matched are collected and acquired in batch-based mode (BBM). Next, an individual sorting model is designed to prioritize tasks and workers by determining the sequence of individuals based on spatiotemporal attributes. Based on the above, the role division model is formulated through the attraction–repulsion mechanism. The algorithm iterates through the two stages of individual sorting and role division. After multiple rounds of information iteration and feedback, the algorithm can obtain the final matching results for task objects.

2 Problem statement

2.1 Scenario description

Based on common real-world service scenarios, we study an HMTA problem from the perspective of SC platforms. The following provides detailed descriptions of real-world scenarios.

2.1.1 Online car-hailing

Online car-hailing is a common application in shared mobility platforms and is also a typical scenario for smart travel. In the online car-hailing service process, users submit travel tasks to the platform online based on their individual requirements. The platform then matches suitable workers (i.e., vehicles) to the users based on the task requirements and spatiotemporal information. Once a match is successful, the worker proceeds to the user's location to provide service. Previous studies have generally treated the travel tasks submitted by users as homogeneous and assumed that all workers have identical attributes. However, in reality, users' travel requirements are diverse, and vehicle types differ. Currently, various shared mobility platforms have provided users with more personalized travel services and car types. The ride-hailing platform DiDi offers users a variety of vehicle types to choose from, including Discount, Express, Flash, Taxi, and Premier (Fig. 1). Users can select different vehicle types based on their travel requirements, considering various factors such as travel cost, waiting time, comfort level, and privacy. Users can also select multiple vehicle types at the same time. Therefore, the tasks published by each user have personalized characteristics, and the platform needs to match one of its vehicles according to the user's choice. In this scenario, the task requirements published by each user are not the same, and the worker attributes differ; it is a typical HMTA problem.

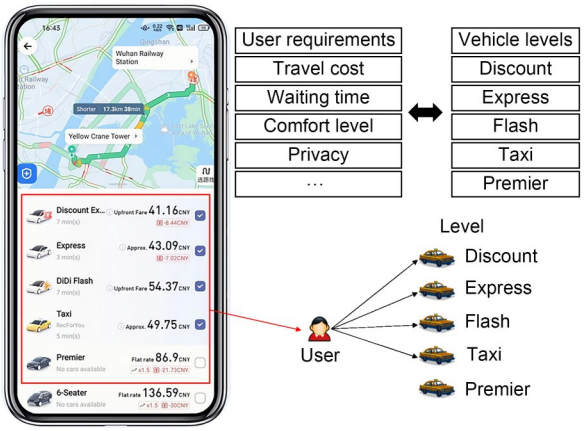

Fig. 1 Different requirements and vehicles in the DiDi car-hailing scenario

2.1.2 On-door repair

On-door repair services have emerged as a new service model based on Internet platforms and sharing economy strategies. Many crowdsourcing repair platforms, such as Shifubang and Jiedantong, have

expanded to cover numerous cities in China. Their service offerings include the installation, repair, and cleaning of various types of furniture and utilities. During the service process, users initiate tasks through the platform and describe the task details. The platform then matches suitable repair workers based on the task requirements and spatiotemporal information to provide on-door services. Importantly, each user's repair task is unique, and the technical skills and qualifications of each worker vary. Therefore, during the task–worker matching process, the platform needs to match workers whose skill levels are equal to or higher than the task requirements to ensure that the task is completed successfully. Thus, this type of on-door repair service is a typical HMTA problem.

Previous research has generally treated tasks posted by users as homogeneous, and all worker types were assumed to be identical. These studies did not consider the individual differences (i.e., heterogeneity) in real-world scenarios. A simple example of a car-hailing scenario is provided in Fig. 2. Over a period of time, users randomly submit a set of tasks to the platform, denoted as $U=\{u_1, u_2, u_3, u_4\}$. Moreover, workers (i.e., vehicles) $W=\{w_1, w_2, w_3, w_4, w_5, w_6, w_7\}$ also appear randomly over time. The spatial distributions of tasks and workers are shown in Fig. 2a, and their temporal distributions are shown in Fig. 2c. The platform offers users five types of vehicles, i.e., Vehicle_type={Discount, Express, Flash, Taxi, Premier}. Users can select different vehicle types based on their travel requirements and choose multiple types simultaneously. For instance, in Fig. 2b, user u_1 selects two vehicle types: Discount and Express; user u_2 selects Express; user u_3 selects Flash and Taxi; user u_4 chooses Express and Flash. The different vehicle type selections by each user result in varied travel tasks, and different types of vehicles have different prices and capacities. During the task allocation process, the platform must fully consider the differentiated characteristics of tasks and workers.

2.2 Problem definition

In response to the described scenario, we define the HMTA problem. A typical SC system includes workers, users, and crowdsourcing platforms. Users post tasks with specific requirements, and workers have various attribute labels. The platform receives all the information from workers and users, allocates tasks to workers based on the allocation mechanism, and earns revenue upon completion of each task. Additionally, the SC is an Internet-based service industry, where service quality is particularly important. In designing the evaluation function, we consider implicit market costs and use the user matching success rate as a measure of user satisfaction. We also focus on both the direct revenue from task allocation and user satisfaction. The goal of the problem is to maximize the overall utility of task allocation while satisfying the constraints. The main symbols are listed in Table 1.

Definition 1 (Task) The crowdsourcing task is defined as a tuple $u_i=<\mathbf{lu}_i, \mathrm{Rl}_i, h_i, s_i, e_i>$, where $\mathbf{lu}_i$ represents the location point of the task u_i in Euclidean space and Rl_i denotes task requirements. The labor cost of task u_i is h_i, indicating the labor (service) cost required by workers to complete the task. The start time of task u_i is s_i, and the end time is e_i. This means that task u_i can be assigned to workers by the platform

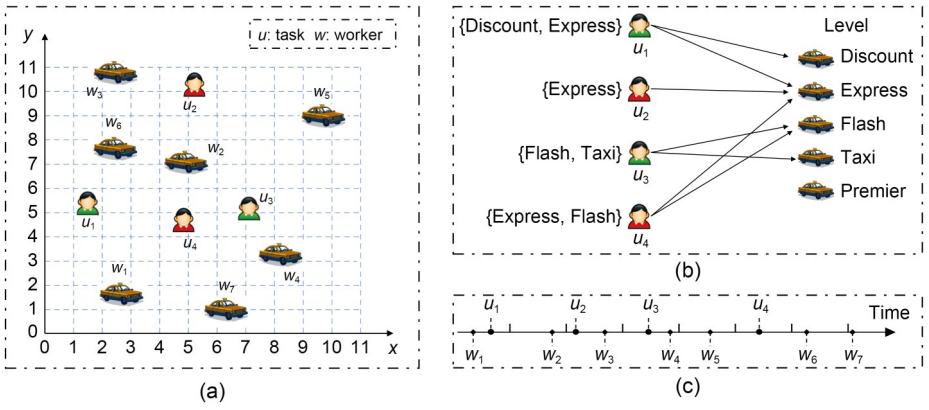

Fig. 2 Task allocation of a car-hailing scenario: (a) spatial distributions; (b) task requirements; (c) temporal distributions

Table 1 Summary of the main notations

Notation	Description
M, M_p	The matching set and the obtained matching set in batch p
U, U_p, Uf	The task set, the collected task set in batch p, and the task set that failed to match in the current batch p
W, W_p	The worker set and the collected worker set in batch p
R, R_i	The total requirement set of tasks and the i^{th} type of requirement
Rl_i	Requirements for task u_i
h_i	The labor cost of task u_i
lu, **lw**	The location of tasks and workers
r_j	The service type provided by worker w_j
pr_j	The unit service price of worker w_j
c_j	The service radius of worker w_j
b_j, d_j	The start and end times of worker w_j
s_i, e_i	The start and end times of task u_i
η, C_d	The dissatisfaction coefficient and unit distance cost coefficient
UT, Ur, Ue	The total utility, direct revenue, and user experience cost of the matching
Ti, Tw	The time interval and the waiting time (the difference between start and end times)
F_{UW}^{ij}	The interaction force between u_i and w_j, which can be obtained through formulas for attraction and repulsion
Rc	The relative coefficient in the role division model
θ	The adjustable threshold of group adjustment strategy

within the time window $[s_i, e_i]$, with the waiting time $\text{Tw}_i=e_i-s_i$.

In Definition 1, the requirements of all tasks can be considered as a set. Taking car-hailing as an example, each user can choose multiple vehicle types for travel. Therefore, the overall requirements of all tasks can be viewed as a set of worker types $R=\{r_1, r_2, \cdots, r_k\}$. The requirements Rl_i for task u_i are described as a subset of the total requirements R, i.e., $\text{Rl}_i \subseteq R$.

Definition 2 (Worker) Crowdsourcing workers are defined as the tuple $w_j=<\mathbf{lw}_j, r_j, p_j, c_j, b_j, d_j>$, where $\mathbf{lw}_j$ represents the current location of worker w_j. The service type of worker w_j is r_j, and the unit service price is pr_j. The start time of worker w_j is b_j, and the end time is d_j. This means that worker w_j can be assigned tasks by the platform within the time window $[b_j, d_j]$, with the waiting time $\text{Tw}_j=d_j-b_j$. The service range of the worker is limited to a circular area centered at the current location $\mathbf{lw}_j$ with a radius of c_j.

In Definition 2, the service type r_j of worker w_j is an element of set R, which can be represented as $r_j \in R$. The platform can assign worker w_j to task u_i only if $r_j \in \text{Rl}_i$. The unit service price pr_j of each worker is related to their service level r_j, with different service levels having different unit prices. U and W represent the sets of spatial tasks and workers, respectively.

Definition 3 (Revenue) The revenue function of worker w_j completing the crowdsourcing task u_i is defined as $\text{Rf}(u_i, w_j)$, which is defined as the difference between the task reward and the execution cost. The specific expression is shown in Eq. (1), where C_d is the unit distance cost coefficient and $\text{dist}(\cdot)$ denotes the distance function. The successful matching between worker w_j and task u_i presupposes that $r_j \in \text{Rl}_i$ and the task is within the service range of the worker.

$$\text{Rf}\left(u_i, w_j\right)=\left[p_j \cdot h_i-C_\text{d} \cdot \text{dist}\left(\mathbf{lw}_j, \mathbf{lu}_i\right)\right] \cdot X_{ij}, \quad (1)$$

$$X_{ij}=\begin{cases}1, & (u_i, w_j) \text{ matching succeeds,}\\ 0, & (u_i, w_j) \text{ matching fails,}\end{cases} \quad (2)$$

$$\text{UT}=\sum\nolimits_{u_i \in U', w_j \in W'} \text{Rf}\left(u_i, w_j\right)-\eta \sum\nolimits_{u_i \in \text{Uf}} h_i. \quad (3)$$

In practical scenarios, crowdsourcing tasks are generally bounded by a specific urban area in terms of space, and the service range of workers is also limited to a certain extent. Therefore, the distance cost in the problem definition is bounded. The service labor cost for each task is also limited to a certain range. Consequently, in practical applications, both prices and profits are bounded. That is, for any task–worker match pair (u_i, w_j), the matching utility has an upper bound B, i.e., $\max(\text{Rf}(\cdot)) \le B$. Based on the above

descriptions, the definition of the HMTA problem can be provided.

Definition 4 (HMTA problem) Given a set of crowdsourcing tasks $U=\{u_1, u_2, \cdots, u_n\}$, a set of workers $W=\{w_1, w_2, \cdots, w_m\}$, and a utility function Rf $(\cdot)$, where tasks and workers appear in an uncertain order, the goal of HMTA is to find a set of assignments M ($M\subseteq U'\times W'$) that maximizes the total utility UT of the task allocation. Here, U' and W' are subsets of task set U and worker set W, respectively. The total utility UT is shown in Eq. (3), where Uf represents the set of tasks that were not successfully assigned and η is the user dissatisfaction coefficient.

When performing task allocation, it is necessary to consider both direct revenue and user satisfaction. Additionally, the matching process must satisfy the following constraints.

1. Time constraint: task allocation can be performed only when both the task and the worker are present. It requires that the activity time $[b_j, d_j]$ of the matched worker w_j intersects with the existence time $[s_i, e_i]$ of task u_i.

2. Spatial constraint: each worker has an activity radius. The prerequisite for a successful match between worker w_j and task u_i is that the task is within the worker's service range.

3. Service constraint: during the allocation process, a match can occur only if the service level r_j of worker w_j can meet the requirements of task u_i (i.e., $r_j\in \mathrm{Rl}_i$).

4. Invariance constraints: once a task allocation is implemented, it cannot be changed.

As described above, the proposed HMTA problem is an online matching problem with bilateral dynamic characteristics. It is a typical combinatorial optimization problem and an NP-hard problem (Zhang et al., 2023). Unlike general bilateral matching problems, the HMTA problem considers individual differences (heterogeneity) and thus has more complex constraints. Exact algorithms are impractical for online problem solving due to their high computational complexity. Many researchers have turned to greedy strategies to address such NP-hard problems (Song et al., 2017a). The main approach involves identifying all potential matching groups and choosing the one that provides the highest benefit (Zhao H et al., 2023). However, the greedy strategy tends to maximize the number of matches, even when the benefits are minimal. While the greedy algorithm offers good efficiency in online matching problems, its performance can be limited by the unpredictable nature of object appearances and the invariance constraints of the problem. As a result, it may sometimes match groups with lower benefits, a limitation that is inherent in real-world applications.

3 Role division approach embedded with an individual sorting model

In this section, we propose RD-ISM to address the HMTA problem. The process begins with data processing of perceived information, where we collect tuples of objects to be matched using a BBM (Fig. 3). Subsequently, we design an individual sorting model to determine the sequence of individuals based on spatiotemporal attributes, thereby prioritizing tasks and workers. The role division model is then developed, consisting of two parts. The first part is a matching strategy based on attraction–repulsion mechanism, which can pair tasks with workers. The second part is a threshold-based heuristic group adjustment strategy, which enables adjustments to matching groups with lower utility. After individual sorting and role division, the current batch matching results are obtained. In the real-time task allocation process, the algorithm can achieve the final matching results of all objects following multiple batches of iteration.

3.1 Task sensing and batch-based mode

Many SC platforms, such as DiDi, Uber, Gigwalk, TaskRabbit, and gMission, are built on the foundation of crowd sensing (Tong et al., 2020). Prior to task allocation, the platform must collect basic information about the objects involved in the allocation (You et al., 2023). In the HMTA problem, the objects fall into two major categories, i.e., tasks and workers, both of which exhibit individual differences; that is, there is heterogeneity.

The HMTA problem proposed in this paper is conducted based on task sensing and is essentially a dynamic matching problem. There are currently two modes for addressing dynamic matching problems. One is the instant assignment mode (IAM), where matching is performed immediately as tasks emerge. The other is the BBM, where tasks and workers are matched after

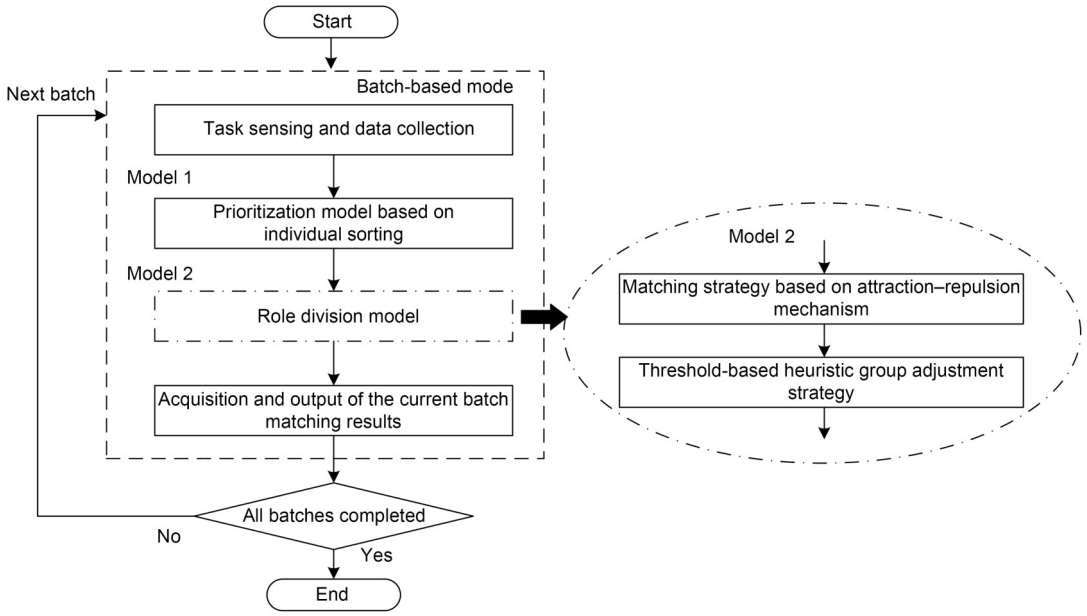

Fig. 3 Overall framework of RD-ISM

a certain time interval, that is, in batches (Liu et al., 2022). Each mode has its own advantages and disadvantages. In IAM, the system executes matching immediately when a new task arrives, resulting in a short task response time. However, due to the limited number of current matching objects, the quality of the matching results is often unsatisfactory. In contrast, BBM performs a unified matching operation for multiple objects that appear within a time interval at fixed moments. In this case, the task response time is slightly extended (related to the time interval), but the quality of the matching results can be significantly improved (Guo et al., 2018). In practical applications, platforms do not strictly enforce immediate assignment, which provides an opportunity to enhance the effectiveness of dynamic task allocation results. For instance, in SC for transportation services like DiDi, the mode used is BBM (Hettiachchi et al., 2022). The process of BBM is shown in Fig. 4. Within a duration time T, a matching operation is conducted after each time interval, with the matching moment denoted as timestamp g. The batch processing procedure is illustrated in Algorithm 1. Specifically, at each timestamp g, the crowdsourcing platform collects a batch of tasks U and workers W. The sources of task set U_p and worker set W_p consist of two parts: first, the new objects that emerge in the current batch; second, the unmatched objects left over from previous batches. During the task sensing process, our proposed method collects information about the objects in batches based on BBM, facilitating subsequent task allocation.

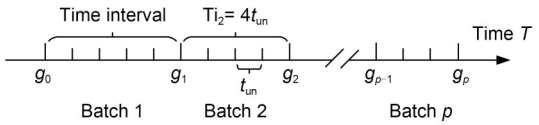

Fig. 4 Timeline diagram of the batch-based mode (BBM)

Algorithm 1 Batch-based mode

Input: Total time T and time interval Ti

Output: Task–worker matching results M within T

Set $M \leftarrow \varnothing$
After every time interval Ti, a batch p is obtained
for each batch $p \in T$ **do**
  Collect a set of tasks U_p
  Collect a set of workers W_p
  Prioritize all objects according to the individual sorting model to obtain a task sequence
  Based on the task sequence, perform matching based on the role division model and obtain a feasible matching $M_p \subseteq U_p \times W_p$
  Let the matching result M_p be a feasible matching of batch p and output it
  Let $M \leftarrow M \cup M_p$
  Adjust the next batch interval Ti according to the changing trend of the number of tasks
end for
Return M

Note that the time interval (the batch size) is not fixed. Taking car-hailing as an example, the platform adjusts the size of the time interval based on traffic flow and task number. Generally, as the number of vehicles and tasks increases, the time interval decreases to facilitate quicker matching. Conversely, when the number of vehicles and tasks decreases, the time interval can increase, provided that it is within the allowable waiting time to ensure the quality of the matching results. In BBM, the number of tasks and vehicles in the upcoming period can be predicted by existing data to determine the next batch size (time interval). This problem can be regarded as a short-term data prediction issue based on time series. Traffic flow data exhibit trends over time, with data points closer to the prediction point having a greater impact on the predicted value. To improve generalization performance and ensure rapid solution capabilities of the proposed method, we refer to the exponential smoothing method and propose a time series model that can adaptively adjust the batch size (Dudek et al., 2022). The main idea is to enhance the role of recent observations in the predicted value, assigning unequal weights to observations at different times, thereby increasing the weight of recent observations and enabling the predicted value to quickly reflect actual market changes.

Assume that during a time period T, the time interval for a batch p is denoted as Ti_p and can be represented as Eq. (4), where C is a positive integer and t_{un} represents the unit time, meaning that the time interval Ti is an integer multiple of the unit time t_{un}.

$$\mathrm{Ti}_p = C \cdot t_{\mathrm{un}}, \tag{4}$$

$$\hat{N}_{t+k} = A_t + B_t k, \tag{5}$$

$$A_t = 2S_t^{(1)} - S_t^{(2)}, \tag{6}$$

$$S_t^{(1)} = \varepsilon N_t + (1-\varepsilon) S_{t-1}^{(1)}, \tag{7}$$

$$S_t^{(2)} = \varepsilon S_t^{(1)} + (1-\varepsilon) S_{t-1}^{(2)}, \tag{8}$$

$$B_t = \frac{\varepsilon}{1-\varepsilon}\left(S_t^{(1)} - S_t^{(2)}\right). \tag{9}$$

Using the unit time t_{un} as the interval, the number of tasks that appear in the $(t+k)^{\mathrm{th}}$ unit time is predicted based on the data from the previous t unit times. The specific formulae are given by Eqs. (5)–(9), where N_t is the number of tasks that appear in the t^{th} unit time and $\hat{N}_{t+k}$ is the predicted number of tasks that appear in the $(t+k)^{\mathrm{th}}$ unit time. A_t represents the intercept term, and B_t represents the slope term in the calculation, both of which are derived from the exponential smoothing values, as shown in Eqs. (6) and (9). ε is the weighting coefficient, where $\varepsilon \in (0,1)$. $S_t^{(1)}$ and $S_t^{(2)}$ are the first-order and second-order exponential smoothing values at time t, respectively.

According to the above formulae, let the time interval of batch p+1 be equal to that of batch p. The predicted number of new tasks emerging in batch p+1 $\widehat{\mathrm{NB}}_{p+1}$ can then be obtained, as shown in Eq. (10). The task quantities of the two batches are compared to decide whether to change the time interval of batch p+1. If the ratio of the task number in batch p+1 to that in batch p is greater than (or less than) a certain threshold, the time interval is adjusted according to Eq. (12), where NB_p is the number of tasks that emerge in batch p. υ_1 and υ_2 are relative coefficients, set as $\upsilon_1 = 1/\upsilon_2 = 1.5$. Additionally, the time interval Ti is an integer multiple of the unit time t_{un} and should be restricted to a certain range. Based on previous work (Feng and Xiao, 2024), it should be less than half of the waiting time of the task objects, i.e., $\mathrm{Ti} \in \left(t_{\mathrm{un}}, 0.5\ \mathrm{Tw}\right)$.

$$\widehat{\mathrm{NB}}_{p+1} = A_t \mathrm{Np} + B_t \sum_{i=1}^{\mathrm{Np}} i, \tag{10}$$

$$\mathrm{Np} = \frac{\mathrm{Ti}_p}{t_{\mathrm{un}}}, \tag{11}$$

$$\mathrm{Ti}_{p+1} = \begin{cases} \mathrm{Ti}_p + t_{\mathrm{un}}, & \widehat{\mathrm{NB}}_{p+1} > \upsilon_1 \mathrm{NB}_p, \\ \mathrm{Ti}_p - t_{\mathrm{un}}, & \widehat{\mathrm{NB}}_{p+1} < \upsilon_2 \mathrm{NB}_p, \\ \mathrm{Ti}_p, & \text{otherwise.} \end{cases} \tag{12}$$

3.2 Priority division model based on individual sorting

Based on task sensing, in this study, we use the individual sorting model to prioritize task objects and determine the task sequence. The individual sorting model is inspired by the spatial structuring observed in the task allocation processes of biological groups. When individuals perform tasks, they exhibit activity levels associated with the tasks, and these activity levels vary among individuals. For instance, in an ant colony, individuals sort themselves within the nest based on activity level. Individuals with lower activity

levels move toward the center of the nest, while those with higher activity levels move toward the periphery. Ultimately, each individual position itself is between those with higher activity levels and those with lower activity levels. The activity area of each individual is confined to a limited space, resulting in an orderly spatial structure for the entire colony. Tasks within the colony are spatially separated, with each individual executing a limited set of tasks within its confined activity area, leading to a clear division. In the process of SC task allocation, worker w needs to execute task u in a timely manner, aiming to complete as many high-reward tasks as possible before the end time to ensure maximum profit.

Based on the idea of individual sorting, task objects can be classified according to the degree of time urgency. For task u and worker w that have not yet matched, their priority functions are represented by Eqs. (13) and (14), respectively, where α, β, and κ represent the waiting time weight, the deadline weight, and the reward weight, respectively, which can be set as follows: $\alpha/\beta<1$ and $\alpha/\beta+\kappa=1$. t_{now} is the current time, and Pr $(\cdot)$ is the priority function.

$$\Pr(u_i)=\alpha(t_{\text{now}}-s_i)/(\beta(e_i-t_{\text{now}}))+\kappa h_i, \quad (13)$$

$$\Pr(w_j)=\alpha(t_{\text{now}}-b_j)/(\beta(d_j-t_{\text{now}})), \quad (14)$$

$$\text{EU}=\left\{u_i\middle|(e_i-t_{\text{now}})/\text{Ti}_{p+1}\leqslant 1,\forall u_i\in U\right\}, \quad (15)$$

$$\text{EW}=\left\{w_j\middle|(d_j-t_{\text{now}})/\text{Ti}_{p+1}\leqslant 1,\forall w_j\in W\right\}. \quad (16)$$

Substitute tasks and workers into the priority function to obtain the values, and then arrange them in descending order to obtain the priority sequences. The larger the Pr $(\cdot)$ value is, the higher the priority. Furthermore, objects that approach their deadlines require special attention. The objects with a remaining time of less than one batch time interval are categorized separately and denoted as sets EU and EW, with their function judgment expressions shown in Eqs. (15) and (16), where Ti_{p+1} is the time interval for the next batch. For objects in sets EU and EW, priority allocation is necessary in the next batch to ensure task completion rates.

3.3 Role division model

Based on the prioritization of objects, a role division model is designed to perform matching operations for task objects. The specific process is divided into two main parts: the attraction–repulsion mechanism-based matching strategy and the threshold-based group adjustment strategy.

The core idea of the attraction–repulsion mechanism is based on individual role differentiation, where individuals (or agents) of different role types undertake different tasks (Xiao et al., 2023). Fig. 5 shows a schematic diagram illustrating the principle of the attraction–repulsion mechanism. During group cooperation, in response to all the tasks present in the environment, each individual (or agent) spontaneously transforms into the execution unit of different roles according to different task demands. Gradually, the whole group shows different forms of organization. In the above process, many individuals participate and perform different tasks. Different tasks have a different attraction to individuals, which is related to the amount of demand (value) of the task. Individuals are also influenced by other participating individuals while choosing their tasks. When the number of individuals participating in a given task is large, other individuals

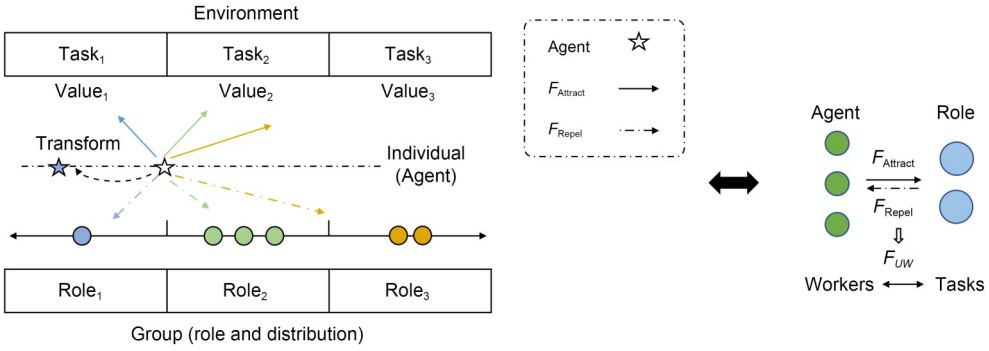

Fig. 5 Attraction–repulsion mechanism and interaction force

are repelled to ensure the average benefit of the participating individual and the team's overall coordination. In this case, a repulsive effect occurs among individuals. Each individual flexibly adjusts his/her role status based on attraction and repulsion, and eventually emerges to obtain the overall role distribution. More theoretical content is provided by Xiao et al. (2023).

3.3.1 Matching strategy

During the task allocation process, worker w needs to travel to the location of task u to provide service, considering spatial and temporal constraints, task requirements, and labor costs. Based on these factors, a mapping relationship between task allocation and the attraction–repulsion mechanism is designed (Table 2). Workers are considered agents of different types, and tasks are regarded as roles. Based on the current environment (the attributes of objects), agents (i.e., workers) undergo role transformation; that is, task–worker matching operations are performed. When the role division of all objects is completed, the overall role distribution (matching result) is obtained. The attraction–repulsion mechanism performs object matching based on the interaction forces between tasks and workers. The interaction forces include two kinds of force: the attractive force F_{Attract} and the repulsive force F_{Repel} (Fig. 5).

Table 2 Mapping relationship between attraction–repulsion mechanism and task allocation

Attraction–repulsion mechanism	Task allocation
Agent	Worker
Role	Task
Environment	Task attributes
Role diversion	Task–worker matching
Overall role distribution	Matching results $M\subseteq U\times W$

In the task allocation process, let $W=\{w_1, w_2, \cdots, w_m\}$ and $U=\{u_1, u_2, \cdots, u_n\}$, representing m workers and n tasks, respectively. The attractive force F^{ij}_{Attract} between worker w_j and task u_i is related to the reward and urgency of the task. The greater the reward and the stronger the urgency are, the greater the attractive force (Eq. (17)), where RT_{ij} is the relative coefficient between task u_i and worker w_j, denoted as $\text{RT}_{ij}=\text{Rn}_j\cdot\text{Rc}_i$.

Rc_i is the relative urgency coefficient of task u_i (Eq. (18)), with a larger Rc_i indicating a higher degree of urgency. Its value range is (0, 0.5). h_i is the labor cost of task u_i, and Rn_j is the task demand coefficient (Eq. (19)). A larger Rn_j indicates lower demand for r_j in the current batch; to ensure the balance of different demands, the task's attractiveness to worker w_j increases accordingly. The set Us_j represents all tasks in the current batch that contain r_j in their task requirements (Eq. (20)).

$$F^{ij}_{\text{Attract}}=\begin{cases}0, & r_j\notin \text{Rl}_i \text{ or } \text{dist}\left(\mathbf{lw}_j, \mathbf{lu}_i\right)>c_j,\\ \text{RT}_{ij}\cdot p_j\cdot h_i, & \text{otherwise},\end{cases} \tag{17}$$

$$\text{Rc}_i=1+\min\left(\frac{\text{Pr}(u_i)}{\max\limits_{u\in U}\left(\text{Pr}(u)\right)}, 0.5\right), \tag{18}$$

$$\text{Rn}_j=\exp\left(1-\frac{\left|\text{Us}_j\right|}{|U|}\right), \tag{19}$$

$$\text{Us}_j=\left\{u_i\middle| u_i\in U \text{ and } r_j\in\text{Rl}_i\right\}, \tag{20}$$

$$F^{ij}_{\text{Repel}}=\text{RT}_{ij}\cdot C_{\text{d}}\cdot\text{dist}\left(\mathbf{lw}_j, \mathbf{lu}_i\right), \tag{21}$$

$$F^{ij}_{UW}=\max\left(\left(F^{ij}_{\text{Attract}}-F^{ij}_{\text{Repel}}\right), 0\right). \tag{22}$$

The repulsive force F^{ij}_{Repel} between task u_i and worker w_j is related to the distance cost. The farther the distance between the task and the worker is, the stronger the repulsive force (Eq. (21)).

The attractive and repulsive forces between each task and the worker can be calculated using Eqs. (17)–(21). The total force F^{ij}_{UW} between task u_i and worker w_j is shown in Eq. (22), and the value of F^{ij}_{UW} is greater than or equal to 0. If the attractive force F^{ij}_{Attract} is greater than the repulsive force F^{ij}_{Repel}, the magnitude of F^{ij}_{UW} is the difference between the two. If the attractive force F^{ij}_{Attract} is smaller than the repulsive force F^{ij}_{Repel}, matching cannot yield any revenue, so the total force F^{ij}_{UW} is 0.

Using the above matching processes, an initial matching result M' between task u and worker w is obtained. The main idea is to regard workers as agents and tasks as roles, and under the interaction of attractive and repulsive forces, different tasks are assigned to workers.

3.3.2 Group adjustment stage

In BBM, the matching result of each batch is related to the degree of matching of objects in the current batch. However, during this process, there are

inevitably some undesirable matching groups with low utility. In such cases, if a matching with low utility is carried out, it may lead to the loss of subsequent matchings with higher utility. To address this challenge, we propose a threshold-based group adjustment strategy. In this strategy, an adjustable threshold θ is designed. It is used to evaluate matching groups (i.e., (u, w)) at the end of the preliminary matching in each batch. We filter the groups whose utilities are less than the threshold θ to eliminate poor-quality matching. This strategy may result in the loss of some allocations with low utility. However, it can select the larger ones among the conflicting allocations and avoid the worst-case scenario. In the process of group adjustment, the key issue is how to determine the appropriate threshold θ when filtering matching groups in each batch. Therefore, a heuristic rule is designed with reference to an existing adaptive thresholding algorithm (ATA) (Song et al., 2017a).

The profit function $\mathrm{Rf}(u_i, w_j)$ defined in HMTA is used as the evaluation metric for the matching group (u_i, w_j) to calculate the value of each group. The expression for $\mathrm{Rf}(u_i, w_j)$ is presented in Eq. (1).

$$\mathrm{Fi}(u_i, w_j) = \begin{cases} 1, & \mathrm{Rf}(u_i, w_j) \geq \theta, \\ 0, & \text{otherwise}, \end{cases} \tag{23}$$

$$M_p = \left\{ (u_i, w_j) \,\middle|\, (u_i, w_j) \in M_p' \text{ and } \mathrm{Fi}(u_i, w_j) = 1 \right\}, \tag{24}$$

$$P_z = \omega_z \Big/ \sum_{z=0}^{v} \omega_z, \tag{25}$$

$$\omega_z = \omega_z (1 + \lambda E_z), z = 0, 1, \cdots, v, \tag{26}$$

$$E_z = \sum_{(u,w) \in M_p^z} \mathrm{Rf}(u, w). \tag{27}$$

The filtering function $\mathrm{Fi}(\cdot)$ is defined in Eq. (23). When the evaluation function $\mathrm{Rf}(\cdot)$ of a matching group is less than the threshold θ, the group is filtered out. At this point, a filtering operation can be performed on each matching group to obtain the final matching results M_p for the current batch p, as shown in Eq. (24). When the algorithm selects different thresholds in a given batch, it leads to different matching results. The proposed group adjustment strategy can dynamically adjust the probability of threshold selection. The group adjustment strategy is applied throughout the entire phase of the RD-ISM algorithm. The overall pseudo-code of the RD-ISM algorithm is shown in Algorithm 2. The maximum evaluation function value $\mathrm{Rf}_{\mathrm{Max}}$ can be set or obtained from the history of the crowdsourcing platform before task allocation.

Algorithm 2 RD-ISM

Input: Total time T and initial time interval Ti

Output: Matching results M

1 Set $M \leftarrow \varnothing$
2 Let $v \leftarrow \mathrm{floor}(\ln(\mathrm{Rf}_{\mathrm{Max}}))$
3 Set ω_z=1, where z=0, 1, ⋯, v
4 Calculate the probability $\boldsymbol{P}=(P_0, P_1, \cdots, P_v)$ according to Eq. (25)
5 After every time interval Ti, a batch p is obtained
6 **for** each $p \in T$ **do**
7 Collect the task set U_p and the worker set W_p
8 Calculate the priority Pr(·) of objects and EU and EW according to the individual sorting model
9 Prioritize the execution of the matching strategy for the objects in EU and EW
10 Execute the matching strategy for the other objects, and integrate to obtain the preliminary matching results M_p' for batch p
11 Set $M_p \leftarrow \varnothing$
12 **for** each (u, w) in M_p' **do**
13 According to probability $\boldsymbol{P}$, choose a random value ve from {0, 1, ⋯, v}
14 Let $\theta \leftarrow \mathrm{e}^{\mathrm{ve}}$
15 Calculate the values of Fi for each matching group (u, w) in M_p' according to Eq. (23)
16 **if** $\mathrm{Fi}(u_i, w_j)$=1 **then**
17 $M_p \leftarrow M_p \cup \{(u_i, w_j)\}$
18 **end if**
19 Update the weights $\{\omega_0, \omega_1, \cdots, \omega_v\}$ according to Eq. (26)
20 Update the probability $\boldsymbol{P}$ according to Eq. (25)
21 **end for**
22 $M \leftarrow M \cup M_p$
23 Adjust the next batch interval Ti according to the changing trend of the task number
24 **end for**
25 Return M

Lines 1–4 are the initializations of the algorithm. First, the parameter v and the set of thresholds $\{\mathrm{e}^0, \mathrm{e}^1, \cdots, \mathrm{e}^v\}$ are calculated, and all the weights $\{\omega_0, \omega_1, \cdots, \omega_v\}$ of the available thresholds are initialized to 1. The selection probability of the available thresholds is calculated by Eq. (25). Lines 5–7 represent the task sensing process under the current batch p. Line 8 prioritizes all the objects based on the individual

sorting model. Lines 9 and 10 represent the preliminary matching results M_p' obtained using the matching strategy in the current batch p. Lines 11–18 select a value e^{ve} from the available thresholds $\{e^0, e^1, \cdots, e^v\}$ based on the selection probability $\boldsymbol{P}$ and evaluate and filter each matching group in the preliminary result to obtain the final matching result M_p in the current batch p. Subsequently, a parameter updating process (lines 19 and 20), i.e., updating the weights ω_z and probabilities P_z in each batch, is used. E_z can be calculated by Eq. (27), which indicates that when e^z is always used as the threshold θ from the first batch, the evaluation sum of the matching results of the current batch p is obtained. The learning rate λ is used to control the update rate of ω. In this study, we set λ= 0.01 according to Song et al. (2017b). The final matching results M of all batches are obtained in lines 22–25.

3.4 Algorithm analysis

The RD-ISM algorithm encompasses two distinct models: the individual sorting model and the role division model (Fig. 3). The individual sorting model, inspired by spatial structuring in biological group task allocation processes, determines the task sequence through the analysis of spatiotemporal attributes. It can ensure the timeliness of task allocation and the task completion rate. According to the priority classification, the role division model allocates each task to the optimal extent through batch cyclic processing and information feedback. Consequently, the proposed RD-ISM ensures the total utility of task allocation while improving the success rate of task matching. Moreover, the approach presented in this paper is implemented in a BBM. Considering practical application scenarios, an adaptive adjustment method for batch time intervals is provided in conjunction with the exponential smoothing model, further enhancing the applicability of the algorithm.

Furthermore, we analyze the computational complexity of RD-ISM. We assume that the task allocation process contains a total of N batches. The time interval between batches needs to be predicted before each batch begins. Our proposed prediction model is obtained based on the improved exponential smoothing method, and the complexity is consistent with that of the exponential smoothing method: both are $O(1)$. Therefore, the complexity of predicting N batches is $O(N)$. In any batch p containing task objects U_p and W_p, the algorithm is divided into two parts (i.e., individual sorting and role division). In the individual sorting model, the tasks U_p and workers W_p are sorted first, and its time complexity is $O(|U_p|+|W_p|)$. In the role division model, there are two stages: the matching stage and the group adjustment stage. In the matching stage, the tasks U_p and workers W_p are matched, and its time complexity is $O(|U_p||W_p|)$, which represents calculating the interaction force. In the group adjustment stage, it is necessary to evaluate each group of the preliminary result M_p' with complexity $O(v|M_p'|)$. In the worst case, $|M_p'|$ equals $|U_p|$, and the complexity is $O(v|U_p|)$. Therefore, in batch p, the complexity is $O(|U_p|+|W_p|+|U_p||W_p|+v|U_p|)$. Assuming the same number of task objects in each batch, the complexity of the entire matching process is O $(N(|U_p|+|W_p|+|U_p||W_p|+v|U_p|)+N)$. In the worst case, there is only one batch process, i.e., N=1. At this point, the complexity takes the maximum value of $O(|U|+|W|+|U||W|+v|U|)$.

4 Experiment

In this section, we apply the proposed RD-ISM to solve the HMTA problem. To evaluate the performance of RD-ISM, we design two distinct sets of test cases: synthetic dataset instances and real-world dataset instances.

4.1 Experimental setup

4.1.1 Datasets

HMTA is a common task allocation problem in SC platforms that considers the differences in task requirements and worker types. For instance, in the car-hailing scenario, tasks posted by users can encompass various vehicle types. Consequently, in this experiment, we set up a total of four different task requirements and worker levels R. Moreover, workers of different levels have different unit service prices pr. The specific parameter settings are shown in Table 3.

The real-world dataset used in this experiment was taxi data from Chengdu in August 2014, which includes the vehicle ID, longitude and latitude, passenger capacity, date, and time for each taxi (Feng and Xiao, 2024). We selected a portion of the data at a ratio of 1:4 to

Table 3 Dataset statistics and parameter settings

Parameter	Settings and statistics	
	Real-world dataset	Synthetic dataset
N=4|U|=|W|	2500, **5000**, 10 000, 20 000	5000, 10 000, **20 000**, 40 000, 80 000
Time range (min)	[0, 300]	[0, 500]
Spatial range	Latitude and longitude to coordinates	Randomly distributed within 30 km×30 km area
Task requirements and worker levels R	{1, 2, 3, 4}	{1, 2, 3, 4}
Unit service price pr (CNY)	{1, 2, 3, 4}	{1, 2, 3, 4}
Worker service radius c (km)	10	5, **10**, 15, 20
Unit distance cost C_d (CNY/km)	0.8	0.8
Dissatisfaction coefficient η	0.1	0.1
Expected value μ of task service cost h	20, **25**, 30, 35, 40	20, **25**, 30, 35, 40

Bold entries represent the default settings used during the experiment

represent tasks (users) and workers (vehicles) (i.e., $|U|:|W|$=1:4). Additionally, missing information (such as unit service price, task service requirements, and unit cost) was supplemented for each dataset according to the definition of the HMTA problem. The statistical information of the real-world dataset is presented in Table 3. The selected time range was from 7:30 am to 12:30 pm (with 1 min as the unit time, totaling 300 min). The location information of the objects could be converted through longitude and latitude, and the price was quoted in RMB. In the experiment, the unit distance cost C_d for workers was set to 0.8 CNY/km, based on real-world conditions, and the service radius c of all workers was set to 10 km. By default, workers of different levels (types) were randomly generated, and each task randomly generated a service requirement set Rl. The task service (labor) cost h defaulted to a normal distribution N(25, 25), with an expected value μ of 25 and a standard deviation σ of 5. The initial time interval Ti for the real-world dataset experiment was set to 5 min, meaning one batch was 5 min. The waiting time Tw of the objects obeyed a uniform random distribution with a distribution interval of [20 min, 30 min]. The grouping information for the dataset experiments is also shown in Table 3.

Furthermore, we tested the scalability of the algorithm using synthetic datasets. Task objects (tasks U, workers W) were generated within a 30 km×30 km area, with their location information following a uniform random distribution. The time range for task allocation was set to [0, 500 min]. Workers of different service levels were randomly generated. Each task randomly generated its service requirement set Rl. The ratio of tasks to workers was 1:4. The task service cost h defaulted to a normal distribution N(25, 25), with an expected value μ of 25 and a standard deviation σ of 5. The waiting time Tw of the objects also followed a uniform distribution with a distribution interval of [20 min, 30 min], and the initial time interval Ti was 5 min. By adjusting the service cost, task number, and other parameters, the impact of various parameters on the algorithm's performance was verified. The details of the synthesized dataset are shown in Table 3.

4.1.2 Compared approaches

To evaluate the performance of RD-ISM, we introduced three algorithms as comparison benchmarks: the greedy algorithm (Tong et al., 2021), delay matching (DM) algorithm (Li et al., 2020), and ATA (Song et al., 2017b). Note that during the experimental process, these comparison algorithms needed to be modified to adapt to the HMTA problem.

Greedy algorithm: The greedy algorithm is a classic algorithm that has been proven to be effective in matching problems (Tong et al., 2021). In performing the greedy algorithm, whenever a new task arrives, the algorithm traverses all available workers and performs matching for the new task. Then, the algorithm finds all the matching groups that satisfy the problem constraints and selects the group with the highest revenue as the matching result. We modified the greedy algorithm to make it suitable for HMTA. We modified the evaluation function of the algorithm as Eq. (1) to maximize the matching revenue and added the service

constraints and spatial constraints during algorithm execution.

DM algorithm: This algorithm adopts the idea of delayed matching, proposed by Li et al. (2020). The tasks are processed when the waiting time is about to end. Essentially, the DM algorithm is similar to the greedy algorithm, but improved by the introduction of delayed matching. As with the greedy algorithm, we modified the evaluation function of the original DM as Eq. (1) and added the service and spatial constraints.

ATA: This algorithm is from Song et al. (2017b) and is an extension of the random threshold algorithm. The algorithm can match groups in real time when objects emerge and update the threshold to filter the groups with high revenue. To apply the algorithm to the HMTA problem, we modified the evaluation function to Eq. (1) and the objective function to Eq. (3).

To ensure the fairness of the experimental process, all the experiments were conducted using a Windows 10 PC with Intel® Core™ i7-12700 2.1 GHz CPU and 32 GB main memory. The algorithms were implemented based on MATLAB R2020a. The criteria used to compare the above algorithms were as follows:

1. Total utility (UT): the total utility of the task allocation, calculated from the HMTA definition.

2. Task completion rate ($|M|/|U|$): the ratio of the matching number $|M|$ to the task number $|U|$.

3. Average time (AT): AT=MT/$|M|$, i.e., the total runtime for task matching MT divided by the number of task matches $|M|$.

4.2 Experiments on the synthetic dataset

First, we analyzed the effect of variation in the number of tasks and workers on the algorithm performance. The other experimental parameters were set to the default values shown in Table 3. The experimental results are shown in Fig. 6a. During the experiment, as the number of workers $|W|$ gradually increased (with $4|U|=|W|$), the total utility UT obtained by the algorithms showed an upward trend. However, the results of the RD-ISM algorithm were always superior to those of the other algorithms, with RD-ISM>ATA>DM>Greedy. When the number of workers was set to the default value (i.e., $|W|$=20000), the total utility UT obtained by RD-ISM was approximately 20% higher than that of ATA, and this gap gradually widened as $|W|$ increased.

As shown in Fig. 6b, when $|W|$=20000, the comparison of the average time AT for each algorithm yielded the result RD-ISM>ATA>DM>Greedy, meaning that the greedy algorithm had the shortest average time AT to complete tasks, followed by the DM algorithm and ATA. The increased average time observed for RD-ISM compared to other algorithms is mainly due to the dynamic adjustment mechanism and individual sorting process. RD-ISM integrates an attraction–repulsion mechanism to dynamically balance task allocation and resource utilization, which inherently involves more computational steps than simpler algorithms like the greedy algorithm. This complexity is necessary to achieve a better balance between task completion and resource efficiency, especially in dynamic environments with many constraints. While the increased computation time may seem like a disadvantage, it is important to note that RD-ISM is designed to optimize long-term efficiency and adaptability rather than purely minimizing execution time. The algorithm's ability to dynamically adjust to changing conditions ensures better coordination and resource utilization, which ultimately leads to higher overall

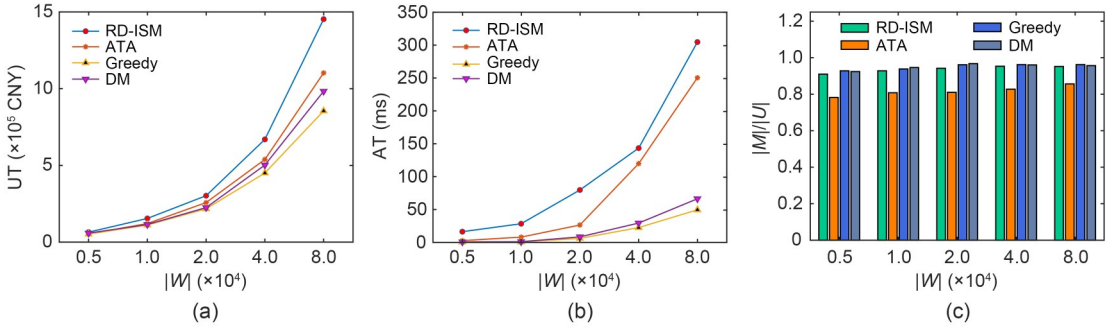

Fig. 6 Comparison of results from varying $|W|$: (a) UT; (b) AT; (c) $|M|/|U|$. DM: delay matching algorithm; Greedy: greedy algorithm. References to color refer to the online version of this figure

system efficiency and profit in various scenarios. The experimental results revealed that the average time AT of RD-ISM was indeed higher than that of the comparison algorithms. However, the time remained within the same order of magnitude, and all average times were within the millisecond range. The difference in runtime decreased as the number of task objects increased (Fig. 6b). As the number of tasks increased, the average time AT for each task also slightly increased. This is because as $|W|$ increases, the number of objects that need to be processed in each batch increases, leading to a greater number of possible matching groups that need to be considered. Additionally, comparison of the task completion rate $|M|/|U|$ of each algorithm shows that RD-ISM was also relatively high (Fig. 6c), slightly inferior to the greedy algorithm, and can meet user experience requirements.

In this experiment, the task service cost h defaulted to a normal distribution $N(25, 25)$. During the experiment, to analyze the impact of the task service cost h on the algorithm performance, the expected value μ of the normal distribution was varied within the set {20, 25, 30, 35, 40}. The utility of RD-ISM was superior to that of the other algorithms (Fig. 7a). As the value of μ increased, the total utility UT of all algorithms increased. The reason is that with increasing expectations, the overall task service cost also increased. Additionally, as μ increased, there was no significant pattern in the average time AT of the algorithms (Fig. 7b). Furthermore, a comparison of the task completion rate $|M|/|U|$ of each algorithm in Fig. 7c shows that the proposed algorithm RD-ISM was also good, and there was no significant pattern with the change in μ.

Subsequently, we investigated the impact of the worker service radius c on the algorithm performance. Each worker had spatial constraints, manifested as their matching objects not exceeding the area covered by their service radius. The service radius was set to {5 km, 10 km, 15 km, 20 km} (Table 3). The experimental results are shown in Fig. 8. As the radius c increased, the number of tasks each worker could match also increased, leading to a slight improvement in the overall task-completion rate $|M|/|U|$ and total utility UT.

The proposed RD-ISM was designed based on the BBM, and an adaptive time interval adjustment strategy is presented in this paper. To further investigate the impact of parameter settings on the results, we examined the task allocation performance of the RD-ISM algorithm when a fixed time interval was used; i.e., we compared the solution effects under different fixed time intervals Ti. The results are shown in Fig. 9. As the time interval Ti increased, the total utility UT of the proposed algorithm first increased and then decreased. With an increasing time interval, the perception process of each batch of tasks can obtain more object information, which helps in global matching and thus improves the matching utility to a certain extent. However, when the time interval was too large, some objects were ignored due to excessive waiting time, leading to a decrease in total utility. Moreover, the longer the time interval is, the more objects needed to be matched in each batch, increasing the computational load and reducing task matching efficiency.

4.3 Experiments on a real-world dataset

In the experiments using the real-world dataset, RD-ISM was also compared with the three other algorithms. The statistical information of the real-world dataset and the model parameter information are shown in Table 3. Comparisons and analyses were conducted

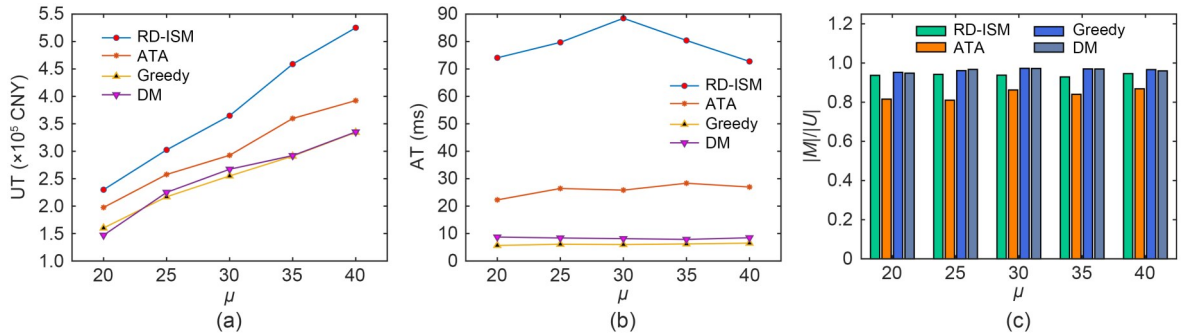

Fig. 7 Comparison of results from varying h (changing the μ of the normal distribution): (a) UT; (b) AT; (c) $|M|/|U|$. References to color refer to the online version of this figure

from three aspects: total utility UT, average time AT, and task completion rate $|M|/|U|$. The comparative results under different scales are shown in Fig. 10. As the number of tasks $|U|$ and the number of workers $|W|$ changed continuously, RD-ISM consistently outperformed the other three algorithms in terms of the total utility UT for task allocation. Moreover, as the scale increased, the total utility obtained by the algorithm increased. The average time AT of RD-ISM was longer than that of the other algorithms (Fig. 10b) but was still within an acceptable range and met the requirement for real-time performance. In addition,

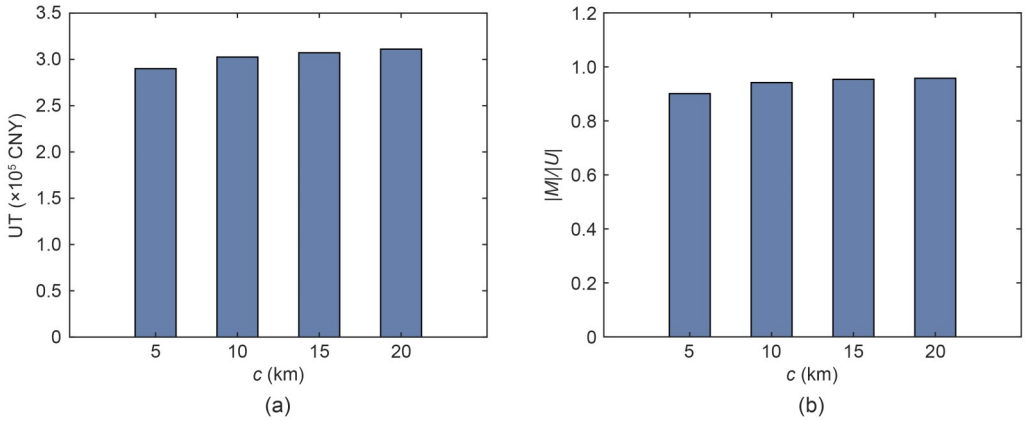

Fig. 8 Comparison of results from varying c: (a) UT; (b) $|M|/|U|$

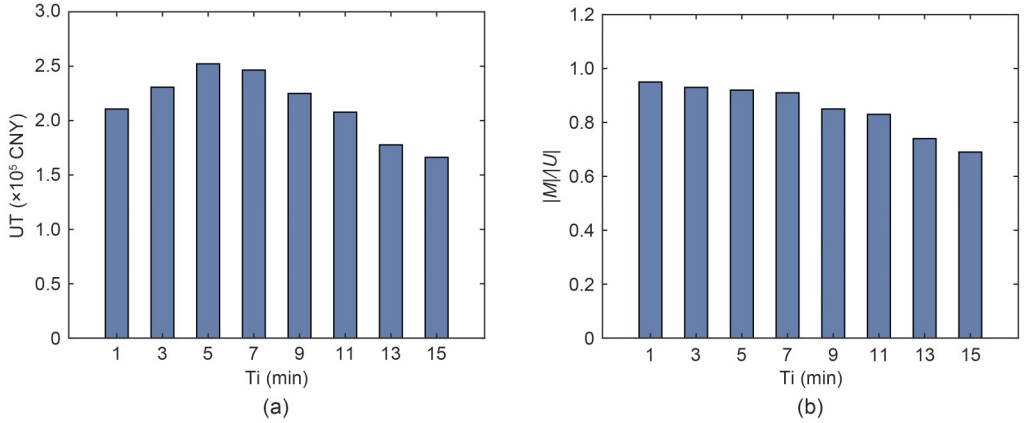

Fig. 9 Comparison of results from varying Ti: (a) UT; (b) $|M|/|U|$

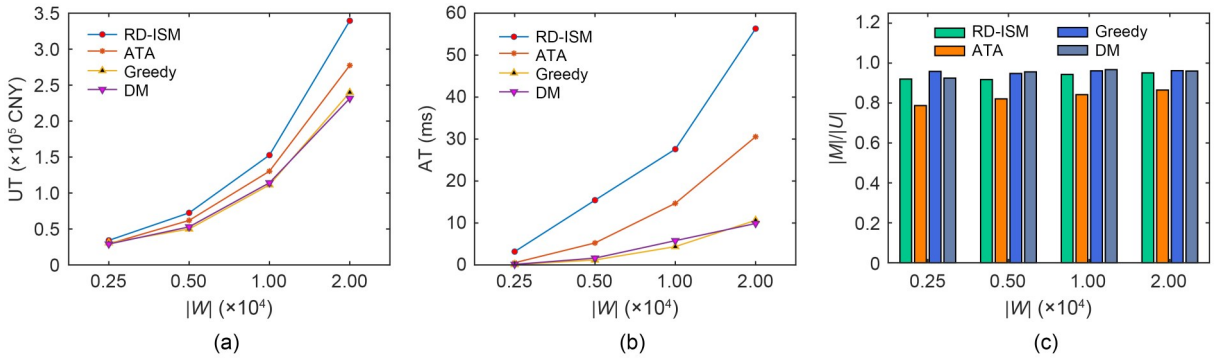

Fig. 10 Comparison of results from varying $|W|$: (a) UT; (b) AT; (c) $|M|/|U|$. References to color refer to the online version of this figure

the task completion rate $|M|/|U|$ of RD-ISM was relatively good, meeting the user experience requirements.

As with the synthetic dataset, to analyze the impact of the service cost h on the algorithm performance during the experiment, the experimental results with different expected values μ were compared. The total utility of RD-ISM was superior to that of the other algorithms (Fig. 11). As the value of μ gradually increased, the total utility UT of all algorithms increased. The reason is that with increasing expected value, the overall task reward also increases. Additionally, as the value of μ gradually increased, the average time AT of the algorithms showed no clear pattern of change.

4.4 Summary of experiments

HMTA problem has strong uncertainty and is difficult to solve precisely. In this study, we designed various tests based on real and synthetic datasets for algorithm verification. In all instances, the effectiveness of RD-ISM in solving the HMTA problem was verified through comparisons with three different algorithms. Additionally, we investigated the impact of different parameter values on performance and validated the scalability of the method. In real-world dataset instances, the effectiveness of the algorithm was verified through experiments at three different scales. Compared with the other algorithms, RD-ISM achieved a better total utility UT in all the experiments. RD-ISM also achieved a satisfactory task completion rate $|M|/|U|$ and average time AT. RD-ISM can ensure a higher task completion rate $|M|/|U|$ while maximizing the total utility UT. Overall, RD-ISM demonstrates good performance in solving the HMTA problem. This method, which is based on task perception, performs rapid matching through individual sorting and role division models and then adjusts the matches using group adjustment strategies to ultimately obtain better matching results. The superiority of RD-ISM lies in its ability to optimize the total utility, which is a more comprehensive measure of system performance. This makes RD-ISM a robust and versatile solution for SC platforms, where balancing multiple objectives is critical for long-term efficiency and adaptability. Experiments also prove that RD-ISM is suitable for solving large-scale online heterogeneous task allocation problems.

5 Conclusions

With the rapid development of mobile Internet technology and the online-to-offline business mode, SC has been widely applied in many industries. In this study, we investigate a class of online HMTA problems in SC. We provide a definition and formal description of the problem based on real-world scenarios. The objective of the HMTA problem is to match task objects (user–worker pairs) in a timely and dynamic manner to maximize the total utility. In HMTA, the task completion rate is used as a measure of user satisfaction, and the utility function is defined by integrating matching direct revenue and market implicit costs. In terms of problem solving, we propose RD-ISM. During the real-time task allocation process, the individual sorting model is first introduced to determine the sequence of individuals based on spatiotemporal attributes, prioritizing tasks and workers. Then, a role division model is designed to match tasks and workers. Moreover, our approach is implemented within a BBM.

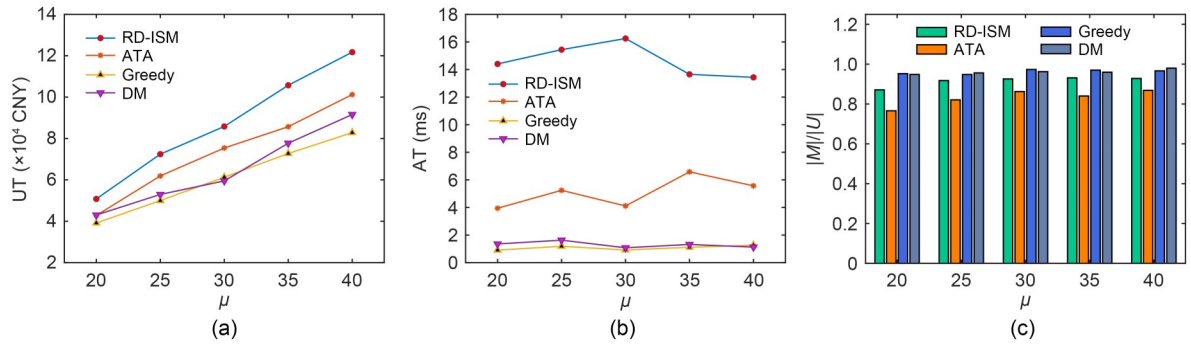

Fig. 11 Comparison of results from varying h (changing the μ of the normal distribution): (a) UT; (b) AT; (c) $|M|/|U|$. References to color refer to the online version of this figure

Considering real scenarios, an adaptive adjustment method for batch time intervals is provided in conjunction with an exponential smoothing model, further enhancing the applicability of the algorithm. Experiments are designed on real-world and synthetic datasets. Through comparisons with other algorithms, the effectiveness and superiority of the proposed algorithm were verified.

While this paper addresses a class of HMTA problems from the perspective of SC platforms and achieves good results, the approach still has limitations in practical applications. Taking traffic traveling as an example, travel services in SC are diverse, and the current problem definition struggles to encompass all problem types, such as task allocation problems involving multi-targets or personal preferences. In some cases, the platform may need to consider the balance of the user experience, and bottleneck matching is another issue that requires further research. Additionally, although the proposed approach is a quasi-real-time online algorithm under dynamic scenarios, it still has limitations. In the algorithm proposed in this paper, the matching process considers factors such as distance costs, but in practical applications, the correlation of the cost function with traffic conditions and sudden accidents is equally important. Future work should consider further improving the algorithm by introducing predictive methods and the concept of stochastic programming.

Contributors

Renbin XIAO designed the research and revised and finalized the paper. Zhenhui FENG proposed the approach, performed the experiments, and drafted the paper. Mingzhi XIAO provided essential support by assisting with computations and verifications. All authors read and approved the final paper.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Alamri S, 2024. The geospatial crowd: emerging trends and challenges in crowdsourced spatial analytics. *ISPRS Int J Geo-Inform*, 13(6):168. https://doi.org/10.3390/ijgi13060168

Bhatti SS, Fan JH, Wang KR, et al., 2021. An approximation algorithm for bounded task assignment problem in spatial crowdsourcing. *IEEE Trans Mob Comput*, 20(8):2536-2549. https://doi.org/10.1109/TMC.2020.2984380

Cai JH, Jia LS, Hu XQ, 2023. Operation decision model in a platform ecosystem for car-sharing service. *Electron Commer Res Appl*, 59:101262. https://doi.org/10.1016/j.elerap.2023.101262

Chen Z, Fu R, Zhao ZY, et al., 2014. gMission: a general spatial crowdsourcing platform. *Proc VLDB Endow*, 7(13):1629-1632. https://doi.org/10.14778/2733004.2733047

Dortheimer J, 2022. Collective intelligence in design crowdsourcing. *Mathematics*, 10(4):539. https://doi.org/10.3390/math10040539

Duan ZJ, Li W, Zheng X, et al., 2019. Mutual-preference driven truthful auction mechanism in mobile crowdsensing. Proc IEEE 39th Int Conf on Distributed Computing Systems, p.1233-1242. https://doi.org/10.1109/ICDCS.2019.00124

Dudek G, Pełka P, Smyl S, 2022. A hybrid residual dilated LSTM and exponential smoothing model for midterm electric load forecasting. *IEEE Trans Neur Netw Learn Syst*, 33(7):2879-2891. https://doi.org/10.1109/TNNLS.2020.3046629

Feng ZH, Xiao RB, 2023. Spatiotemporal distance embedded hybrid ant colony algorithm for a kind of vehicle routing problem with constraints. *Front Inform Technol Electron Eng*, 24(7):1062-1079. https://doi.org/10.1631/FITEE.2200585

Feng ZH, Xiao RB, 2024. Three-dimensional task allocation for smart transportation in spatial crowdsourcing: an intelligent role division approach. *Adv Eng Inform*, 62:102736. https://doi.org/10.1016/j.aei.2024.102736

Gong DW, Peng C, Yao XJ, et al., 2020. A model of new workers' accurate acceptance of tasks using capable sensing. *Swarm Evol Comput*, 59:100732. https://doi.org/10.1016/j.swevo.2020.100732

Guo B, Liu Y, Wang LY, et al., 2018. Task allocation in spatial crowdsourcing: current state and future directions. *IEEE Int Things J*, 5(3):1749-1764. https://doi.org/10.1109/JIOT.2018.2815982

Hassan UU, Curry E, 2014. A multi-armed bandit approach to online spatial task assignment. Proc IEEE 11th Int Conf on Ubiquitous Intelligence and Computing and IEEE 11th Int Conf on Autonomic and Trusted Computing and IEEE 14th Int Conf on Scalable Computing and Communications and Its Associated Workshops, p.212-219. https://doi.org/10.1109/UIC-ATC-ScalCom.2014.68

Hettiachchi D, Kostakos V, Goncalves J, 2022. A survey on task assignment in crowdsourcing. *ACM Comput Surv*, 55(3):49. https://doi.org/10.1145/3494522

Li BY, Cheng YR, Wang GR, et al., 2020. 3D-online stable matching problem for new spatial crowdsourcing platforms. *J Softw*, 31(12):3836-3851 (in Chinese). https://doi.org/10.13328/j.cnki.jos.005969

Liang L, Fu JD, Zhu HB, et al., 2023. Solving the team allocation problem in crowdsourcing via group multirole assignment. *IEEE Trans Comput Soc Syst*, 10(3):843-854.

https://doi.org/10.1109/TCSS.2022.3155868

Lin XC, Wei KM, Li ZT, et al., 2024. Aggregation-based dual heterogeneous task allocation in spatial crowdsourcing. *Front Comput Sci*, 18(6):186605. https://doi.org/10.1007/s11704-023-3133-6

Liu Z, Li KL, Zhou X, et al., 2022. Multi-stage complex task assignment in spatial crowdsourcing. *Inform Sci*, 586:119-139. https://doi.org/10.1016/j.ins.2021.11.084

Mazzetto S, 2024. A review of urban digital twins integration, challenges, and future directions in smart city development. *Sustainability*, 16(19):8337. https://doi.org/10.3390/su16198337

Ray A, Chowdhury C, Bhattacharya S, et al., 2023. A survey of mobile crowdsensing and crowdsourcing strategies for smart mobile device users. *CCF Trans Pervas Comput Interact*, 5(1):98-123. https://doi.org/10.1007/s42486-022-00110-9

Song TS, Tong YX, Wang LB, et al., 2017a. Online task assignment for three types of objects under spatial crowdsourcing environment. *J Softw*, 28(3):611-630 (in Chinese). https://doi.org/10.13328/j.cnki.jos.005166

Song TS, Tong YX, Wang LB, et al., 2017b. Trichromatic online matching in real-time spatial crowdsourcing. Proc IEEE 33rd Int Conf on Data Engineering, p.1009-1020. https://doi.org/10.1109/ICDE.2017.147

Tong YX, Yuan Y, Cheng YR, et al., 2017. Survey on spatio-temporal crowdsourced data management techniques. *J Softw*, 28(1):35-58 (in Chinese). https://doi.org/10.13328/j.cnki.jos.005140

Tong YX, Zhou ZM, Zeng YX, et al., 2020. Spatial crowdsourcing: a survey. *VLDB J*, 29(1):217-250. https://doi.org/10.1007/s00778-019-00568-7

Tong YX, Zeng YX, Ding BL, et al., 2021. Two-sided online micro-task assignment in spatial crowdsourcing. *IEEE Trans Knowl Data Eng*, 33(5):2295-2309. https://doi.org/10.1109/TKDE.2019.2948863

Wang L, Yu ZW, Han Q, et al., 2018. Multi-objective optimization based allocation of heterogeneous spatial crowdsourcing tasks. *IEEE Trans Mob Comput*, 17(7):1637-1650. https://doi.org/10.1109/TMC.2017.2771259

Wang MZ, Wang YJ, Sai AMVV, et al., 2022. Task assignment for hybrid scenarios in spatial crowdsourcing: a Q-Learning-based approach. *Appl Soft Comput*, 131:109749. https://doi.org/10.1016/j.asoc.2022.109749

Wang YJ, Cai ZP, Zhan ZH, et al., 2019. An optimization and auction-based incentive mechanism to maximize social welfare for mobile crowdsourcing. *IEEE Trans Comput Soc Syst*, 6(3):414-429. https://doi.org/10.1109/TCSS.2019.2907059

Wu HS, Li H, Xiao RB, et al., 2018. Modeling and simulation of dynamic ant colony's labor division for task allocation of UAV swarm. *Phys A Stat Mech Appl*, 491:127-141. https://doi.org/10.1016/j.physa.2017.08.094

Xiao RB, 2024. Four development stages of collective intelligence. *Front Inform Technol Electron Eng*, 25(7):903-916. https://doi.org/10.1631/FITEE.2300459

Xiao RB, Feng ZH, Wu BW, 2023. Research on emergence mechanism of collective intelligence from the complexity perspective. *Int J Bio-Inspir Comput*, 22(1):28-39. https://doi.org/10.1504/IJBIC.2023.133500

You JP, Jiang HW, Chen ZY, et al., 2023. Order allocation strategy for online car-hailing platform in the context of multi-party interests. *Adv Eng Inform*, 57:102110. https://doi.org/10.1016/j.aei.2023.102110

Zhang Q, Wang YJ, Cai ZP, et al., 2022. Multi-stage online task assignment driven by offline data under spatio-temporal crowdsourcing. *Dig Commun Netw*, 8(4):516-530. https://doi.org/10.1016/j.dcan.2021.10.005

Zhang Q, Wang YJ, Yin GS, et al., 2023. Two-stage bilateral online priority assignment in spatio-temporal crowdsourcing. *IEEE Trans Serv Comput*, 16(3):2267-2282. https://doi.org/10.1109/TSC.2022.3197676

Zhao BM, Xu P, Shi YX, et al., 2019. Preference-aware task assignment in on-demand taxi dispatching: an online stable matching approach. Proc 33rd AAAI Conf on Artificial Intelligence, p.2245-2252. https://doi.org/10.1609/aaai.v33i01.33012245

Zhao H, Pan QK, Gao KZ, 2023. A cooperative population-based iterated greedy algorithm for distributed permutation flowshop group scheduling problem. *Eng Appl Artif Intell*, 125:106750. https://doi.org/10.1016/j.engappai.2023.106750

Zhou JJ, Gao L, Lu C, 2023. Solving multi-task manufacturing cloud service allocation problems via bee colony optimizer with transfer learning. *Adv Eng Inform*, 56:101984. https://doi.org/10.1016/j.aei.2023.101984