Expert Systems With Applications 254 (2024) 124442

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Selecting workers like expert for crowdsourcing by integration evaluation of individual and collaborative abilities

Yaohui Han [a], Mingyang Zhao [b], Nuanqiao Shan [c], Anfeng Liu [a,*], Tian Wang [d], Houbing Song [e], Shaobo Zhang [f]

[a] School of Electronic Information, Central South University, Changsha 410083, China
[b] School of Computer Science and Engineering, Central South University, Changsha 410083, China
[c] School of Computer Science, Fudan University, Shanghai 200433, China
[d] Artificial Intelligence and Future Networks, Beijing Normal University & UIC, Zhuhai 519087, China
[e] Department of Information Systems, University of Maryland, Baltimore County (UMBC), Baltimore, MD 21250, USA
[f] School of Computer Science and Engineering of the Hunan University of Science and Technology, Xiangtan 411201, China

ARTICLE INFO

Keywords:
Worker selection
Mobile crowdsourcing
Shapley value
Team formation
Multi-armed bandit

ABSTRACT

Team-based worker selection has been extensively studied for Mobile Crowdsourcing (MCS), in which a set of workers are recruited to form a team to complete complex tasks collaboratively. However, existing studies face two typical challenges: 1) how to dynamically evaluate workers' individual abilities and collaborative contributions to the team; 2) how to select unknown workers to form a team with high quality at low cost. To tackle the above challenges, this paper proposes an Integration of Individual and Collaborative Abilities based Dynamic Worker Selection (IICA-DWS) algorithm to recruit excellent workers as a team in a high-quality and low-cost style. In the IICA-DWS algorithm, each worker's individual ability and collaborative contribution to the team are evaluated more accurately using the Approximate Shapley Value (ASV). In addition, a high-quality team formation method is established to complete complex tasks at low cost. This involves the selection of both team leaders and team members. In this process, the Multi-Armed Bandit (MAB) model is adopted to dynamically select excellent workers using exploration and exploitation phases. Lastly, the IICA-DWS algorithm is evaluated through theoretical analysis and experimental results. The results show that the IICA-DWS algorithm can improve the data quality of tasks by 47.3% and reduce the cost by 61.7% on average. Moreover, the IICA-DWS algorithm has a high probability of approximating optimal results, which shows the best performance among the comparative algorithms.

1. Introduction

MCS has emerged as one of the most important solutions to complete tasks (Campana & Delmastro, 2022; Al-qaness et al., 2022; Nguyen & Zeadally, 2021; Bai et al., 2023). In MCS, workers carrying sensing devices with abundant computational power (Wang, Liu, et al., 2024; Fu et al., 2024; Lu et al., 2024;) can complete multiple tasks, such as data-collecting tasks (Tang, Han, et al., 2023; Peng et al., 2024; Yu et al., 2023). In these tasks, workers collect data in a certain area and report it to the platform, which has powerful data analysis and processing ability (Ouyang et al., 2023; Yang, Zeng, et al., 2023; Tang, Fan, et al., 2023). The platform then can process the data and construct it into various applications (or services) to complete the tasks (Wang, Liu, et al., 2024; Ouyang et al., 2023; Xu et al., 2022). The tasks mentioned above can be classified as simple tasks which do not show any relevance. Workers do not have to collaborate in these cases.

There are many MCS applications, such as Aircloud for air quality measurement (Cheng et al., 2014), Sensorly (Sensorly, 2021) for constructing cellular/WiFi network coverage maps, Nericell (Mohan et al., 2008), and VTrak (Thiagarajan et al., 2009) for traffic detection, and Geograph (Geograph, 2020) for street view collection.

Due to the developments in micro-processing technology, the computational power of many sensing devices, such as smartphones, far exceeds that of computers from over a decade ago. Additionally,

* Corresponding author.
E-mail addresses: han0129@csu.edu.cn (Y. Han), 8208210118@csu.edu.cn (M. Zhao), 21307130395@m.fudan.edu.cn (N. Shan), afengliu@mail.csu.edu.cn (A. Liu), tianwang@bnu.edu.cn (T. Wang), h.song@ieee.org (H. Song), shaobozhang@hnust.edu.cn (S. Zhang).

https://doi.org/10.1016/j.eswa.2024.124442
Received 25 January 2024; Received in revised form 5 May 2024; Accepted 5 June 2024
Available online 12 June 2024

complex tasks can be accomplished by the collaboration of numerous workers (Zhang et al., 2022), such as monitoring moving targets (Wang, Yang, et al., 2022).

Nowadays, there are some MCS platforms oriented to completing complex tasks, such as Upwork[1] (Jiang et al., 2022; Yin et al., 2023; Zhao et al., 2022). However, most of the proposed MCS studies are focused on simple tasks (Nguyen & Zeadally, 2021; Lu et al., 2024), and collaboration is unnecessary in this situation (Tang, Han, et al., 2023; Peng et al., 2024). In such studies, the criteria for selecting workers primarily revolves around the quality of task completion. These studies require a holistic consideration of issues such as the cost of completing tasks and the completion rate (Peng et al., 2024; Tang, Fan, et al., 2023).

Most current research on worker recruitment for simple tasks in MCS is based on the following assumption: the quality and cost of recruiting workers satisfy submodular and linear functions, respectively (Karaliopoulos, Telelis, et al., 2015; Wang, Yang, et al., 2017). The meaning of task completion quality satisfying a submodular function primarily entails two aspects: Firstly, it has monotonicity. That is, as the number of elements in the set increases, the gain produced by the set does not decrease. In other words, the more workers are recruited, the higher the quality of applications is, or at least it does not decrease. Secondly, it satisfies a set function with diminishing marginal benefits. When selecting workers for an application, as the number of workers increases, the rate at which the application quality improves gradually decreases. The cost of completing tasks satisfies a linear function. It means that the cost of recruiting workers is directly proportional to the number of recruited workers (Karaliopoulos, Telelis, et al., 2015; Wang, Yang, et al., 2017).

In subsequent studies, there has been an increasing focus within MCS on handling more complex tasks. These tasks are composed of multiple sub-tasks. In some cases, such as workflow, sub-tasks have strict execution orders. The output of the previous task may serve as the input for the next. For these tasks, some studies also adopt the approach of selecting high-quality workers for each task individually. In other words, these studies treat complex tasks as simple ones. However, in such cases, the task completion quality does not necessarily satisfy a submodular function, and the cost may not necessarily follow a linear function. The reason for this is the issue of collaboration among workers in complex tasks. Suppose there is good collaboration among workers who are assigned to upstream and downstream tasks; even if the individual abilities of the workers are not high, the completion of tasks will be smoother, and the overall data quality of tasks will be increased. Conversely, suppose there is poor collaboration among workers. In that case, even if the individual abilities of the workers are high, the lack of solid collaboration among them will still result in low-quality task completion.

For example, in the task of assigning several workers to monitor moving targets in different areas jointly, if there is good collaboration, the previous worker will notify the next worker of the characteristics and trajectories of the moving targets, allowing seamless integration between tasks and achieving high completion quality (Wang, Yang, et al., 2022). By contrast, weak collaboration among the workers may lead to a long delay in monitoring the tracked targets and low monitoring quality. In this case, the completion quality for complex tasks does not satisfy submodular functions. This means that increasing the number of workers does not necessarily improve target monitoring quality. Additionally, the cost does not necessarily follow a linear cost function because an increase in the number of workers may lead to an increase in the cost of coordination among them, resulting in a steeper increase rather than a linear function.

To this end, team formation has been explored recently (Jiang et al., 2022; Liao et al., 2021), in which the requester recruits workers with different qualities or skills to form a team to complete complex tasks collaboratively (Wang, Yang, et al., 2022; Jiang et al., 2022; Yin et al., 2023; Zhao et al., 2022, Lykourentzou et al., 2016, Pan et al., 2016; Wang, Jiang, et al., 2016; Liao et al., 2021). There has been some research on Team-based Workers Recruitment (TWR) issues designed for MCS (Jiang et al., 2022; Liu, Luo, et al., 2015; Fathian et al., 2017). Even if in simple tasks, recruiting proper workers is already a challenging issue. In TWR issues, there are more factors, such as worker recruitment cost, quality, and task completion rate. These factors need to be considered simultaneously, making TWR issues multi-objective optimization problems (Nguyen & Zeadally, 2021; Lu et al., 2023; Tang, Han, et al., 2023; Peng et al., 2024; Liu et al., 2019). These issues are even more challenging because the collaboration among workers should be considered (Jiang et al., 2022; Yin et al., 2023; Zhao et al., 2022, Lykourentzou et al., 2016, Pan et al., 2016; Wang, Jiang, et al., 2016). There have been some studies on TWR where two main factors are considered during worker recruitment: (1) prioritizing workers with high individual abilities (Jiang et al., 2022; Zhao et al., 2022); (2) prioritizing workers with strong collaborative abilities (Jiang et al., 2022; Zhao et al., 2022).

There are various research methods on TWR issues. One of them involves the requester's recruiting workers for each sub-task and forming a team composed of all the recruited workers to complete the general complex task (Jiang et al., 2022; Zhao et al., 2022). However, in this approach, the team is formed by the requesters themselves, and it is difficult for the requesters to evaluate the collaboration among the workers, making it challenging to ensure effective collaboration. The stress in these methods lies in acquiring and evaluating information about the collaboration among workers. In Wu et al.'s study (Wu et al., 2023), the collaboration among workers is measured by the frequency of their social interactions. If the frequency of interactions among workers is high, the workers are considered to have strong collaboration and are accorded priority in the selection process (Wu et al., 2023). However, strong interactions among workers do not necessarily indicate strong collaboration. Because, in practice, some interactions may be conflict-driven or adversarial rather than cooperative. Therefore, this team formation approach fails to accurately reflect the inherent collaboration among workers, leading to less optimal results.

Some studies propose a two-level recruitment method with a team leader (Xu et al., 2022). The main task of the requester is to select the team leader, and then the team leader selects the team members. The advantage of this method is that the team is formed by the team leader, who has more information about the workers' collaborative abilities than the requester, potentially resulting in a team with stronger collaboration. However, the limitation of this approach is that it relies on the selection of the team leader. If a poor team leader is chosen, the overall quality of the team may suffer. Additionally, the platform cannot select team members, making it challenging to evaluate workers accurately. In previous research on MCS, methods such as leaderless teams or workers in the team uploading collected data to the edge server are involved (Gad-Elrab et al., 2022). These methods significantly improve the efficiency of the MCS platform in collecting and managing data. However, none of the existing methods can take advantage of collaboration within teams to improve platform efficiency. Therefore, the study of TWR still faces significant challenges, especially in the following aspects:

(1) One of the most challenging issues is how to evaluate the individual and collaborative abilities of workers within the team. In TWR, the team leaders and team members should possess the following characteristics: individual strong task completion abilities (Nguyen & Zeadally, 2021; Lu et al., 2023), and strong collaboration with other team members (Xu et al., 2022; Wang, Yang, et al., 2022; Jiang et al., 2022). In addition to these factors, team leaders are also expected to have strong social influence (Wu et al., 2023) to increase the probability of forming a strongly collaborative team (Wu et al., 2023). However, it is challenging

[1] https://https://www.upwork.com/, accessed August 1, 2021.

to evaluate the workers' abilities in complex tasks accurately. The main reason is that high-quality completion of complex tasks may not necessarily indicate workers' strong individual abilities but could result from strong collaboration among workers.

(2) Another challenging issue is selecting unknown workers to form a high-quality team at low cost. Recruiting workers for simple tasks in MCS is already problematic. Due to the enormous number of workers, the requester can only acquire information about a small portion of workers' qualities (Tang, Han, et al., 2023; Gao et al., 2021). Therefore, if the requester always selects high-quality workers from the known pool, the selection result would be only locally optimal (Tang, Han, et al., 2023; Gao et al., 2021). This is because the requester is unaware of the existence of higher-quality workers in the unknown pool. In TWR research, not only are there numerous workers with unknown qualities, but also the number of teams with unknown qualities formed by different combinations of workers far exceeds the number of individual workers. This significantly increases the complexity of the TWR issues in complex tasks compared to that in simple tasks. Therefore, one of the unresolved issues in worker recruitment is how to gradually approach optimal results by continuous exploration to ensure that the system's task completion quality improves over time.

To tackle the above challenges, in this paper, the IICA-DWS algorithm is proposed to recruit excellent workers as a team to complete complex tasks collaboratively with high quality and low cost. Compared to previous research, we introduced a Shapley value-based method to the IICA-DWS algorithm to evaluate workers' individual and collaborative abilities. This allows for a more accurate identification of workers' contributions. Based on this, an MAB-based model of team formation is proposed, which involves a two-stage construction process for both the team leaders and team members. This dynamic approach enables the performance of complex task completion to approach optimal results dynamically. In summary, the main innovations of this work are as follows:

(1) An Integration of Individual and Collaborative Abilities Evaluation (IICAE) approach is proposed to evaluate the comprehensive abilities of workers more accurately. Because of the high complexity of the traditional Shapley value approach, we have made improvements to make it applicable for evaluating workers' contributions in MCS. Compared with previous evaluation methods, the most significant difference is that the IICAE approach evaluates workers' collaboration in practice rather than the frequency of workers' social interactions. Thus, it is reasonable to evaluate the results in practice during the subsequent worker selection process.

(2) A worker selection method is established to build high-quality teams at low cost. Workers with strong comprehensive abilities and social influence are selected as team leaders, contributing to the formation of high-quality teams. Workers with high efficiency are then selected as team members. What sets this approach apart from previous research is that all selections are based on the IICA-DWS algorithm proposed in this paper.

(3) Lastly, we adopted an MAB-based model to choose optimized workers by using the exploration and exploitation phases comprehensively. In most TWR solutions, worker selection was only based on known workers' qualities, which lacked the ability to approach optimal results. However, the MAB-based model adopted in this work enables the IICA-DWS algorithm to achieve better performance. The IICA-DWS algorithm is evaluated through theoretical analysis and experimental results. The results demonstrate that the IICA-DWS algorithm can average an improvement in task completion quality by 47.3 %. Additionally, it can simultaneously reduce the cost by 61.7 % and minimize the cumulative regret by 27.3 % on average.

The rest of this paper is organized as follows. In Section 2, the related works are reviewed. The system model and research objective are presented in Section 3. In Section 4, the IICA-DWS algorithm is introduced in detail. Then, Section 5 provides the performance evaluation. Finally, the conclusion is given in Section 6.

2. Related works

Recently, MCS has garnered significant attention in industry and academia as its development is thriving (Tian et al., 2020; Karaliopoulos, Bakali, et al., 2020; Cai, Duan, et al., 2020). In MCS, devices with substantial computational resources and advanced sensing capabilities are required. A significant number of workers equipped with such devices are recruited to complete various tasks like data collection, target monitoring, environment surveillance, and diverse computational tasks (Yucel, Yuksel, et al., 2020; Yucel & Bulut, 2020).

In MCS, the tasks are mainly centered around data collection (Liu, Xie, et al., 2024). In such tasks, several workers are recruited to collect data in a specified area (Peng et al., 2024; Tang, Fan, et al., 2023). Most data collection tasks belong to the category of simple tasks, where each worker only needs to collect data from their designated location (Tang, Fan, et al., 2023). In other words, there is no need for collaboration among them.

Some other tasks are referred to as complex tasks, which typically consist of multiple sub-tasks that are interrelated and have dependencies in their execution. To illustrate, the result of one task may become the starting condition for the next task (Jiang et al., 2022; Pan et al., 2016; Wang, Jiang, et al., 2016). An example of such complex tasks is workflow or Directed Acyclic Graph (DAG) tasks (Wang, Yu, et al., 2020). In MCS, such complex tasks include monitoring bird migration; when one worker detects a bird, he or she notifies the next worker of the area where the bird is likely to go, enabling continuous monitoring of the bird (Tang, Fan, et al., 2023). Moreover, due to the significant computational capabilities of the crowds, many DAG tasks can be distributed among workers for completion (Wang, Yu, et al., 2020). The recruitment of workers for complex tasks is much more challenging and has greater practical significance, which is the primary focus of this research.

High task completion quality and low cost are two key issues in worker recruitment within MCS (Tang, Han, et al., 2023; Yang, Zeng, et al., 2023; Ren et al., 2018). However, achieving these two goals can be highly challenging. Firstly, workers invest their time, computational resources, and communication resources to complete tasks (Gao et al., 2021; Yucel, Yuksel, et al., 2020). Thus, they require payments from requesters to motivate their active participation in tasks (Gao et al., 2021; Yucel, Yuksel, et al., 2020). However, choosing high-quality workers is difficult, even for simple tasks. Generally, task completion quality is proportional to the effort invested by the workers, and different workers may possess various skills, resulting in different task completion qualities (Yang, Zeng, et al., 2023; Tang, Fan, et al., 2023). Therefore, worker selection becomes a multi-objective optimization problem (Liu et al., 2019; Ren et al., 2018). In practice, the selection process considers not only task completion quality and cost but also the coverage rate, which refers to the proportion of the monitoring area covered by the collected data.

Under these conditions, research has shown that worker selection remains an NP-complete problem. As a result, several studies have solved this problem with heuristic algorithms to derive approximate solutions (Liu et al., 2019; Ren et al., 2018). The most common approach uses workers' qualities or bids as the selection criterion. This means that workers with higher qualities and lower bids are prioritized. To reduce cost, Liu et al. (Liu et al., 2019) proposed a dynamic reward mechanism. In their approach, for tasks located in the central urban areas where many workers are willing to participate, the platform offers fewer rewards to save system costs. On the other hand, in the less populated outskirts of the city, where there is a lower availability of workers, the platform provides higher rewards to motivate workers to participate in tasks actively (Liu et al., 2019). Adopting such a differential pricing strategy allows high performance and low cost.

However, the worker selection methods in simple tasks often struggle to yield good results in complex tasks. Some complex tasks require strong collaboration among workers while handling upstream and downstream tasks (Wang, Yu, et al., 2020). Even if individual workers possess strong abilities, the overall quality of the complex tasks suffers if there is a lack of collaboration among them (Jiang et al., 2022; Zhao et al., 2022; Lykourentzou et al., 2016). Consequently, researchers have proposed various methods tailored for complex tasks. Initially, Xu et al. (Xu et al., 2022) proposed a worker recruitment method similar to the approach in our work. They aimed to address the issue of requester's inability to recruit enough workers (Xu et al., 2022). They introduced a two-tiered worker recruitment method. In the first tier, requesters select some workers who can recruit more workers to expand the pool because of insufficient workers' willingness to participate (Xu et al., 2022). In terms of rewards, requesters not only provide certain payments to directly recruited workers but also offer additional rewards to those who recruit other workers to incentivize further recruitment contributions (Xu et al., 2022). In their method, the role of directly recruited workers is similar to that of the team leaders in this paper. However, their method primarily aimed at addressing the issue of insufficient platform-recruited workers and did not consider collaboration among them. Therefore, this approach can still be considered a simplification for dealing with complex tasks.

The significant difference between complex and simple tasks lies in the interaction among sub-tasks. Collaboration among workers is essential to achieve high-quality completion of complex tasks (Zhao et al., 2022). Wang et al.'s research (Wang, Yang, et al., 2022), belonging to this category, proposes that the quality of applications constructed by platforms is impacted significantly by the likelihood of worker collaboration rather than solely by the qualities of workers. Therefore, they introduced a strategy for task allocation based on worker collaboration tendencies, affirming the effectiveness of this approach. Some studies have validated the influence of worker collaboration on the quality of applications (Lu et al., 2023; Wang, Yang, et al., 2022; Lykourentzou et al., 2016; Pan et al., 2016; Fathian et al., 2017; Wu et al., 2023). Nevertheless, they also do not provide methods to determine workers' collaboration in practice. Estrada et al. (Estrada et al., 2017) proposed a computing framework for tasks with location and time constraints, using particle swarm optimization technology to select suitable workers based on existing constraints to ensure that the platform can obtain high-quality task results within a given time. How to efficiently manage worker information and collected data in the platform is also an important issue. Yin et al. (Yin, Lu, et al., 2021) equated the problem to a directed maximum spanning tree problem and verified the effectiveness of the method through extensive experiments. However, in their study, the collaboration between workers was not studied and optimized.

To deal with the problems mentioned above, some studies have proposed methods of using the intensity of social interactions among workers as a measure of collaboration (Wang, Jiang, et al., 2016), thereby suggesting how to select workers to form teams to tackle complex tasks collaboratively. Wang et al. (Wang, Jiang, et al., 2016) consider the social interactions among workers as a reliable indicator of efficient team collaboration. Therefore, in such studies, higher social interactions among workers imply stronger collaboration, leading to higher-quality team formation for completing complex tasks (Wu et al., 2023). Their approach closely resembles the methodology outlined in this work. Their method primarily consists of two key steps: the first step involves selecting a team leader, where workers engaging in more social interactions tend to assume leadership roles. The subsequent step involves selecting team members, where those who have strong social interactions are more likely to be chosen (Wu et al., 2023). These methods excel in addressing the issue of how to evaluate actual collaborative abilities between workers compared to earlier research. However, the reality of social interactions does not always align with their assumption. Heightened interactions between workers do not necessarily signify high collaboration; if these interactions are solely work-related, competitive, or even hostile (such as legal disputes), the collaboration among workers might be negative. Genuine intimacy between individuals often does not solely manifest in the intensity of interactions. Therefore, solely depending on the intensity of social interaction to represent worker collaboration can be inaccurate.

Due to the effectiveness of TWR methods in enhancing the quality and reducing cost associated with completing complex tasks, it has garnered significant attention from scholars and researchers (Wang, Yang, et al., 2022; Jiang et al., 2022; Yin et al., 2023; Zhao et al., 2022; Lykourentzou et al., 2016; Pan et al., 2016; Wang, Jiang, et al., 2016). Given the nascent research stage in this field, there has not been a standardized set of terminologies used across various studies. For instance, in Jiang et al.'s study (Jiang et al., 2022), selecting multiple workers to form a batch is essentially identical to recruiting them to form a team in this work. Meanwhile, Yin et al. (Yin et al., 2023) introduce the concept of a Cooperative Unit (CU), in which workers are organized into CUs. In essence, the concept of CU bears similarity to the notion of a team.

At present, some excellent methods for the TWR problem have been proposed. For example, Abououf et al. (Abououf et al., 2019) solved the problem of allocating tasks and workers in MCS by clustering similar tasks and selecting worker groups through genetic algorithms, and using taboo search algorithms to minimize task completion time and total distance traveled by workers, which is very important in the completion of complex tasks. The experimental results show that the Group-based multi-task Worker Selection model can effectively solve the task allocation problem of the platform and greatly improve the efficiency of the platform in completing MCS tasks.

Above all, it is evident that the key to recruiting workers for complex tasks lies in evaluating their collaboration. Using the intensity of social interactions among workers as an indicator of collaboration is merely a preliminary method and may not authentically reflect the true situation. Therefore, this paper proposes an empirical approach. The Shapley value, originating from game theory, addresses the problem of allocating value $v(m)$ created by the cooperation of m individuals (Chen et al., 2022; Xie & Lui, 2022). The data quality of task $Q(m)$ involving m workers can be evaluated, which reflects the collaboration of m workers. Subsequently, the application of the Shapley value method enables the precise calculation of each worker's contribution to the successful completion of the complex task. Workers making significant contributions indicate substantial involvement in completing the complex task. If replacing workers with lesser contributions with a different set improves the quality of completing the complex task $Q(m)$, it implies that the newly introduced workers collaborate more effectively. This method allows for testing and comparing the strengths of collaboration among different workers, guiding the selection of highly collaborative workers to improve the quality of task completion. Moreover, this empirical approach utilizes actual collaboration strengths obtained from workers in the completion of complex tasks. Thus, it can reflect collaboration among workers objectively, which previous methods failed to achieve. Azzam et al. (Azzam, Mizouni, et al., 2018) proposed an excellent group recruitment method, Stable-GRS, and introduced stability as an important criterion for evaluating the MCS platform. While ensuring the effectiveness of the method, it also ensures the stable operation of the platform in continuous complex tasks. In the Stable-GRS method, they introduced the Shapley value to evaluate the actual contribution of participants to data quality, which can be regarded as a reflection of the comprehensive ability of MCS participants. Through this method, the platform will select the most suitable participants to complete the corresponding perception tasks through a greedy algorithm. Experiments based on real-life datasets show that their method has extremely high stability and can effectively guarantee the performance of the platform. However, directly employing the Shapley value method incurs significant complexity, making it challenging to be applied in TWR strategies. Thus, further research in this area is required.

There remains another important issue in TWR that current research has yet to address: how to optimize the selection of workers for better results. Previous studies often focus on selecting high-quality workers based on their known abilities to accomplish tasks. However, this approach merely represents a localized optimization method. It optimizes the selection of workers with already known qualities, leaving out numerous workers with unknown qualities from the pool of eligible selections. While simple tasks have been well studied in terms of dynamic worker selection, methods for complex tasks are notably lacking. MAB models are effective in dynamically selecting workers. In such models, the platform frequently selects high-quality workers from the known pool. However, there is still a probability of selection from the pool of workers with unknown qualities (Tang, Han, et al., 2023; Gao et al., 2021; Cai et al., 2022). If the workers chosen during exploration show high qualities, it enriches the pool of usable workers with high-quality candidates, ultimately enhancing the overall quality of worker selection. However, if the qualities of workers selected during exploration are low, it only impacts the result of the current worker selection. The combined exploration–exploitation approach used in MAB models brings the results closer to optimal outcomes (Tang, Han, et al., 2023; Gao et al., 2021; Cai et al., 2022). Nevertheless, TWR for complex tasks is still in its infancy. One of the key issues this paper aims to address is how to make TWR approximate optimal results for complex tasks.

3. Problem statement

3.1. System model

In this work, the MCS network studied is similar to those in most research, as depicted in Fig. 1. The platform employs workers within a limited budget and enables collaboration among them to accomplish tasks, such as collaborative data collection, thereby optimizing the data quality. Specifically, it can be divided into the following three main components:

Platform: It accepts task requests from the requester, analyzes and assigns them to suitable workers. After the tasks are completed, it collects the data reported by the workers, analyzes and processes it, and then returns the results to the requester.
Workers: They refer to individuals who are equipped with sensing devices or have sensing abilities. They are numerous and can move within a specified area. When they receive tasks, they travel to the destinations and upload the data to earn rewards.
Requester: He or she initiates task requests to the platform, receives data provided by the platform after the task completion, and pays the platform for the cost of collecting and processing data.

Definition 1 (Task, Budget) Assume the platform has received m spatiotemporal sensitive data collection task requests from the requester, defined as $t = \{t_1, t_2, \cdots, t_m\}$. Each task has a fixed budget, and the platform's total cost for recruiting workers to complete tasks cannot exceed it. Specifically, each task has the following characteristics:

1) The total budget for task t_j is B_j.
2) Task t_j requires recruiting at least R_j workers ($R_j \geq 2$).
3) The duration of task t_j is specified as $\left[T_{t_j}^{start}, T_{t_j}^{end}\right]$.
4) The spatial scope covered by task t_j is defined as L_{t_j}.

Definition 2 (Worker, Bid, Team) Assume there are n workers in the system, defined as $W = \{w_1, w_2, \cdots, w_n\}$. They have the following characteristics:

1) Each worker can only engage in one task at a time.
2) Each worker is required to upload his or her social connections to the platform during registration (Yang & Wang, 2015).
3) For worker w_i, the online period is defined as the set $\left[T_{w_i}^{start}, T_{w_i}^{end}\right]$.
4) For worker w_i, the coverage area is defined as the set L_{w_i}.

MCS employs a reverse auction mechanism to recruit workers. The requester and the platform act as buyer and seller, respectively. Products being sold are services for task completion and data collection. After the platform initiates a task, all candidate workers will report their bids to the platform according to the cost required for collecting data. We define the bids of workers for task t_j as $\beta^j = \left\{\beta_1^j, \beta_2^j, \cdots, \beta_n^j\right\}$. Following the result of the reverse auction, the platform selects suitable workers to form several teams, denoted as $g = \{g_1, g_2, \cdots, g_k\}$ based on the bids reported by workers and information from worker profiles. For task t_j, we use x_p^j to denote whether team g_p participates in task t_j: when the value of x_p^j is 1, it indicates that team g_p is involved in completing task t_j. At the same time, all workers in a team can only participate in one current task. Otherwise, it signifies that the team would not participate.

Definition 3 (Ground Truth Data, Sensing Quality) Ground Truth Data (GTD) represents the true and reliable data in a task. In MCS, GTD can be used to evaluate the accuracy of the data collected by the platform. It can also serve as a benchmark to measure the quality of data reported by workers and evaluate their sensing abilities. Due to the influence of the precision of sensing devices and random factors in real scenarios, the data sensed by workers often differs from the GTD. Therefore, to describe the accuracy of workers in collecting data during

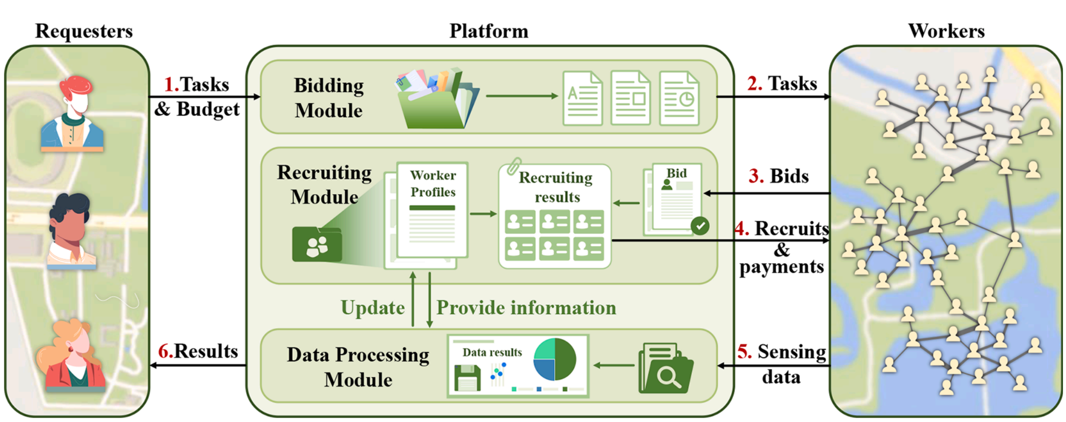


Fig. 1. Workflow of MCS.

task t_j, we define the sensing qualities of workers as $q^j = \left\{q_1^j, q_2^j, \cdots, q_n^j\right\}$. The sensing qualities of workers are associated with both individual and various external factors. Consequently, for worker w_i participating in different tasks t_j and $t_{j'}$, q_i^j may not be equivalent to $q_i^{j'}$.

Sensing quality can describe the accuracy of individual workers in providing data to the platform. In practical situations, workers within a team often engage in collaboration. When workers collaborate effectively, the overall data quality of tasks tends to be higher. Consequently, solely evaluating workers based on sensing qualities while overlooking their collaborative behaviors is inadequate. Such an evaluation system might not maximize the platform's benefits. When collaboration within a team is extremely weak or when workers intentionally act against each other, it can deteriorate the overall data quality of tasks even if individual workers possess strong abilities. To thoroughly investigate workers' collaboration, we define the concepts of collaborative degree and collaborative ability.

Definition 4 (Collaborative Degree, Collaborative Ability) Collaborative degree describes the strength of collaboration among workers while jointly completing tasks. For worker w_i, the collaborative degree with other workers is defined as $\chi_i = \{\chi_{i1}, \chi_{i2}, \cdots, \chi_{in}\}$. The collaborative degree between a worker w_i and himself, namely χ_{ii}, is always 1. Meanwhile, collaborative ability depicts the average collaborative degree of a worker with all other workers. A stronger collaborative ability often implies that the worker can make a larger contribution to the team.

However, collaborative ability is a specific attribute of the workers and is challenging to compute directly. Therefore, in Section 4, we will introduce the IICA-DWS algorithm, which applies the ASV to evaluate the contributions of workers in tasks. This approach allows us to consider workers' collaborative abilities in the evaluation of workers.

Definition 5 (Social Intensity) Social intensity is a quantitative measure of the strength of social connections between two individuals. Within a team, workers with stronger social connections may establish more positive collaboration with a greater number of workers while executing tasks, thereby devoting higher contributions to the team. However, it is challenging to quantitatively evaluate the strength of these social connections solely based on the relationship between two workers in the social network. Hence, we introduce the concept of social intensity. For two workers w_a and w_b in the system having a social connection, their social intensity is defined as:

$$\mathbb{C}_{ab} = \frac{F(w_a) \cap F(w_b)}{F(w_a) \cup F(w_b)} \tag{1}$$

We measure the strength of social connections between two workers using Jaccard similarity (Niwattanakul et al., 2013). Here, $F(w_a)$ represents the friend set of worker w_a. If two workers share more mutual friends, it indicates a stronger social connection. By calculating social intensity, the platform not only confirms the existence of social connections between workers but also evaluates their strengths.

Definition 6 (Social Network) The social network is constituted by aggregating the social connection information uploaded by workers during their registration. As shown in Fig. 2, each worker is represented as a node, with connected nodes indicating the worker's friends. The edge values between nodes represent the social intensity among these friends. Through the social network, we obtain insights into workers' social connections within the MCS system. Based on this network, the platform is enabled to select suitable workers to form teams with strong social connections.

Through the social network, we can determine if there is a social connection between any two workers and the strength of that connection by social intensity. However, there is a problem yet to be solved in this method. Consider a worker with numerous social connections, all of which are relatively weak, versus another worker with fewer connections, but all of them are strong. In such cases, deciding which worker holds a more significant position in the social network becomes

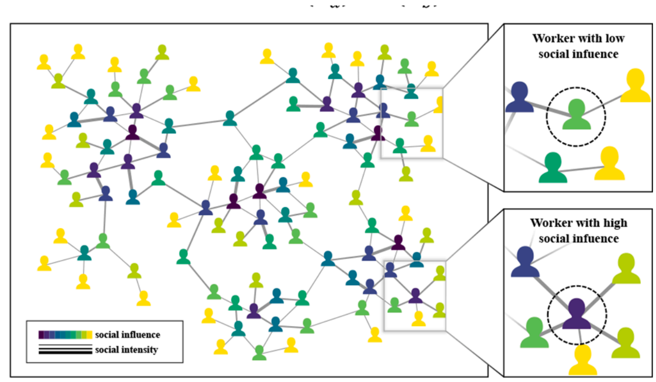


Fig. 2. Social Network.

challenging.

Definition 7 (Social Influence) To solve the problem mentioned above, we introduce social influence to comprehensively evaluate the strength and breadth of social connections. It can indicate an individual worker's significance within the social network. Specifically, the social influence of worker w_a is defined as:

$$\zeta_a = \sum\nolimits_{b=1}^{n} \mathbb{C}_{ab} \cdot y_{ab} \tag{2}$$

Here, y_{ab} denotes whether there exists an edge between w_a and w_b in the social network, indicating the presence (assigned a value of 1) or absence (assigned a value of 0) of a social connection between the two workers.

Definition 8 (Comprehensive Sensing Ability) To evaluate workers' individual and collaborative abilities comprehensively, we define comprehensive sensing ability. After the completion of task t_j, the comprehensive sensing abilities of workers are represented as $\psi^j = \left\{\psi_1^j, \psi_2^j, \cdots, \psi_n^j\right\}$. By computing this value, the platform can have a unified standard to evaluate workers. Simultaneously, to illustrate all workers' comprehensive abilities within a team, we define the team's sensing ability, which is the average of the comprehensive sensing abilities of all workers within the team participating in task t_j:

$$\phi_p^j = \sum\nolimits_{i=1}^{n} \psi_i^j \cdot \nu_i^j \tag{3}$$

Here, ν_i^j denotes whether worker w_i participates in task t_j. If yes, $\nu_i^j = 1$; otherwise, it is 0.

3.2. Research objective

Definition 9 (Platform Efficiency) To better illustrate the platform's ability in collecting data, we define platform efficiency E to describe the performance of the platform in worker evaluation and team formation. Platform efficiency can be represented as follows:

$$E(\phi, B, \beta) = \frac{1}{m} \sum\nolimits_{j=1}^{m} \frac{B_j \cdot \sum\nolimits_{p=1}^{k} \phi_p^j \cdot x_p^j}{\sum\nolimits_{i=1}^{n} \beta_i^j \cdot \nu_i^j} \tag{4}$$

After task t_j is completed, the platform receives the data submitted by all workers who have participated in this task, represented as $d^j = \left\{d_1^j, d_2^j, \cdots, d_n^j\right\}$. Typically, a dataset from a task approximates a Gaussian distribution. The platform aims to increase the data quality while minimizing the payments to workers to maximize its profits. To achieve this goal, based on the available information, the platform must try its best to select efficient workers to participate in tasks. Additionally, total costs and budgets fluctuate due to variations in the difficulty and complexity of different tasks. Therefore, we consider the cost-to-budget ratio as relative cost in Eq. (4), enabling a fair comparison between different tasks.

Our objective is to maximize the comprehensive sensing abilities of recruited workers while minimizing the total cost. In other words, the aim is to maximize platform efficiency under the constraints of budget and the number of workers recruited for tasks. Therefore, we can model the MCS problem addressed in this study as follows:

Maximize:

$$E(\phi, B, \beta) \tag{5}$$

Subject to:

$$\sum\nolimits_{i=1}^{n} v_i^j \geq R_j, \forall t_j \in t \tag{6}$$

$$\sum\nolimits_{i=1}^{n} \beta_i^j \bullet v_i^j \leq B_j, \forall t_j \in t \tag{7}$$

Specifically, the platform needs to recruit a sufficient number of workers to participate in tasks to ensure data accuracy. Eq. (6) denotes that the number of workers recruited for task t_j must exceed the minimum required number of workers R_j. Eq. (7) represents that the total cost of task t_j must not exceed the budget B_j allocated for task t_j. These two constraints enable the platform to avoid issues related to insufficient worker participation or excessively high costs in any tasks, thus preventing a decrease in platform efficiency or a reduction in data quality. The description of major notations in this article are shown in the Table 1.

4. IICA-DWS algorithm

In this section, we introduce the IICA-DWS algorithm. As shown in Fig. 3, it considers both individual and collaborative abilities for worker evaluation and team formation. It iteratively computes the contributions of workers within teams and offers an advanced method for dynamic worker evaluation and selection. This method aims to maximize data quality within a limited budget, in other words, to maximize platform efficiency. Detailed explanations of the IICAE approach will be provided. We will also outline the selection method of team leaders and team members within the IICA-DWS algorithm. Additionally, we will conduct a theoretical analysis of certain properties of the IICA-DWS algorithm.

Table 1
Description of major notations.

Notations	Description
m	Number of tasks
n	Number of workers
B_j	Budget for completing task t_j
β_i^j	Bid of worker w_i participating in task t_j
d_i^j	Data collected by worker w_i in task t_j
q_i^j	Sensing quality of worker w_i in task t_j
χ_i	Collaborative ability of worker w_i
$\mathbb{C}_{ab}$	Social intensity between workers w_a, w_b
ζ_i	Social influence of worker w_i
A^j	The set of workers participating in task t_j
ϕ_p^j	Sensing ability of the team g_p after task t_j
$\widehat{\vartheta}_i^j$	ASV of worker w_i after task t_j
γ_i^j	Regulatory factor of worker w_i after task t_j
h_i^j	RCR of worker w_i after task t_j
h_i^{j+}	UCB index of worker w_i after task t_j
ψ_i^j	Comprehensive sensing ability of worker w_i after task t_j
E	Platform efficiency
p_i^j	The number of tasks that worker w_i has participated in after the task t_j
R_j	The minimum required number of workers
ω_i^j	SV of worker w_i after task t_j

4.1. IICAE approach based on Shapley value

4.1.1. Basic idea

In order to form more efficient teams, the platform needs to evaluate the performance of each worker before making selections. However, individually calculating the precise collaborative abilities of all workers involves high computational complexity and cost. Hence, we devised the ASV to comprehensively assess workers' contributions within teams. According to the characteristics of ASV, we designed the IICAE approach, ensuring a reasonable evaluation of workers throughout the entire process.

In real scenarios, certain tasks may involve multimodal data, with each modality possessing different data characteristics. Therefore, directly comparing data from different modalities is unfair. We require a more detailed evaluation of data across different modalities. Suppose worker w_i participates in task t_j, and the data collected by the worker for this task is represented as d_i^j, comprising multiple modalities. Each dimension of the data corresponds to a modality in the task. For the p-th dimension of data, the platform computes the clustered data $\overline{d}^j(p)$ from the single-dimensional data set $d^j(p)$ collected by all workers participating in this task. Specifically, we cluster the data collected by workers using the Density-based spatial clustering of applications with noise (DBSCAN) method. DBSCAN method is very robust to noise and outliers. In MCS, it enables the platform to set more accurate standards and reduce the impact of outliers on the system. This clustered data is considered as the GTD for that dimension. Following Def. 3, we calculate the worker's sensing quality q_i^j by computing the standard Euclidean distance:

$$q_i^j = \frac{1}{exp\left(\sqrt{\frac{1}{P}\sum\nolimits_{p=1}^{P}\frac{\left[d_i^j(p) - \overline{d}^j(p)\right]^2}{s_j(p)^2}}\right)} \tag{8}$$

Here, $s_j(p)$ represents the standard deviation of the p-th dimension of data. Through this design, we can uniformly assess data from different modalities, ensuring fairness in the platform's evaluation of workers.

Within the IICAE approach, the platform comprehensively evaluates both the individual and collaborative abilities of workers. We now focus on explaining how to leverage workers' collaborative abilities, assisting the platform with a more detailed and rational evaluation of workers to improve platform efficiency further.

Firstly, to visually illustrate how collaboration within a team influences the platform's strategies, we may consider a simplified scenario: Suppose the platform assigns a task and recruits three workers, w_a, w_b, and w_c, to form a team. Assume w_a, w_b, and w_c possess similar individual sensing qualities, and their respective collaborative abilities are $\chi_a = 1.0, \chi_b = 1.5, \chi_c = 0.7$. From the perspective of collaborative abilities, worker w_b positively contributes to the team's sensing ability, while worker w_c has a negative impact. If a worker with higher collaborative ability replaces worker w_c, the team's efficiency might be potentially improved. However, in the scenarios when the collaboration is weaker, replacing worker w_c might decrease the team's efficiency since worker w_c has higher individual sensing quality $\check{q}_c$. In such cases, the platform faces challenges in making decisions.

Furthermore, in reality, not every two workers within a team necessarily exhibit significant collaborative relationships. Also, as the effects of collaboration manifest in workers' individual sensing qualities, precisely computing these collaborative relationships would be extremely challenging. The extensive computation and task results required may render the method impractical within MCS.

The analysis above indicates that it is not advisable to evaluate individual and collaborative abilities separately. Therefore, we need to find a more comprehensive evaluation method.

Shapley (Shapley, 1953) proposed fundamental requirements for the

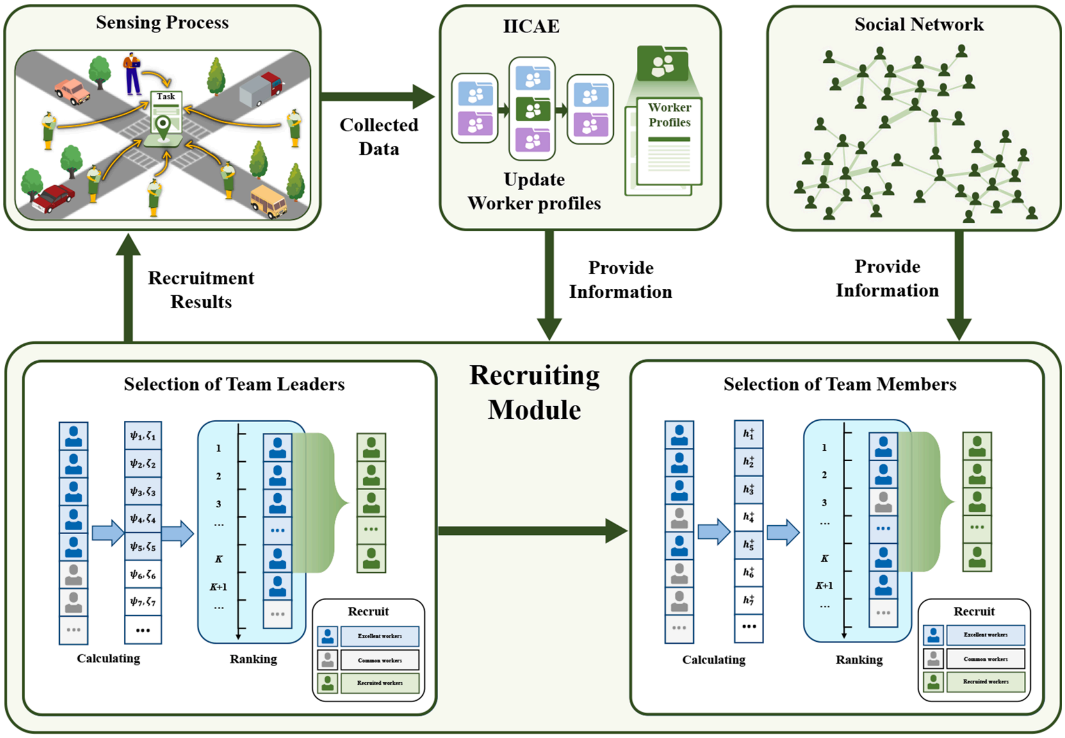


Fig. 3. The framework of IICA-DWS.

fair allocation of team profits, represented by four main properties: balance, symmetry, additivity, and zero element. Balance demands that the team profits should be fairly distributed among all team members. Symmetry dictates that members with equal marginal contributions should receive equal rewards. Additivity implies that the sum of rewards for two tasks should be the sum of individual task rewards. Zero element stipulates that if a member makes no marginal contribution, he or she should not receive any reward.

Based on the statements above, the Shapley value can assist the platform in understanding the contribution of individual workers within this team while ensuring fairness to meet our requirements. Considering all workers $W = \{w_1, w_2, \cdots, w_n\}$ within the same team, let the set S be a subset of all workers completing the task except w_i. The function $Q(S)$ represents the sensing quality the subset S exhibits. Hence, the marginal contribution of an individual worker w_i to the subset S is $Q(S\cup\{w_i\}) - Q(S)$. The Shapley value computes the marginal contribution of an individual worker w_i in all possible worker combinations:

$$\theta_i = \frac{1}{|W|}\sum\nolimits_{S\subseteq W\backslash\{w_i\}} \frac{Q(S\cup\{w_i\}) - Q(S)}{\binom{|W|-1}{|S|}} \tag{9}$$

From Eq. (9) above, it is evident that if the platform needs to compute the Shapley value for a particular worker w_i, it should enumerate all subsets S and conduct extensive and repetitive computations of the sensing qualities exhibited by these subsets. Most importantly, not every worker participates in a certain task. Hence, directly computing the Shapley value using Eq. (9) in the MCS problem is impractical. Additionally, existing research (Chalkiadakis, 2022) has demonstrated that the problem of deriving the Shapley value of a team is NP-hard. Therefore, we suggest a cost-effective approach to estimate each worker's contribution.

Definition 10 (Partial Shapley Value) In task t_j, the Partial Shapley Value (PSV) of worker w_i is defined as:

$$\omega_i^j = \frac{1}{|A^j|}\sum\nolimits_{S\subseteq A^j\backslash\{w_i\}} \frac{Q(S\cup\{w_i\}) - Q(S)}{\binom{|A^j|-1}{|S|}} \tag{10}$$

In this equation, A^j represents the entire set of workers participating in task t_j. ω_i^j computes the total marginal contribution of worker w_i in task t_j relative to the whole set of workers involved in this specific task. Compared to the traditional Shapley value, it confines the computation process within the current team, significantly reducing computational complexity.

However, in practical applications, different tasks vary in difficulty and are influenced by complex unknown factors. This discrepancy makes direct comparison of PSVs across different tasks unfeasible. Therefore, we normalize the PSVs for each task with a min–max normalization to mitigate the impact of this factor. Min-max normalization maintains the relative values of workers' PSVs within a certain task, making it more suitable for evaluating the relative contributions of individual workers across different tasks.

Definition 11 (Normalized Partial Shapley Value) For a given worker $w_i(w_i \in A^j)$ with PSV ω_i^j, we define his or her Normalized Partial Shapley Value (NPSV) in task t_j as:

$$N\left(\omega_i^j\right) = \frac{\omega_i^j - min(\omega^j)}{max(\omega^j) - min(\omega^j)} \tag{11}$$

It is important to note that individually computing NPSVs for all workers will incur significant computational costs. Thus, based on the characteristics of PSV, we have devised a more efficient method for computing NPSV. Suppose two workers, w_a and w_b, participating in task t_j. The difference in their PSVs can be calculated as follows:

$$\Delta\omega_{a,b}^{j} = \frac{1}{|A^{j}|}\sum_{S\subseteq A^{j}\setminus\{w_a\}}\frac{Q(S\cup\{w_a\})-Q(S)}{\binom{|A^{j}|-1}{|S|}} - \sum_{S\subseteq A^{j}\setminus\{w_b\}}\frac{Q(S\cup\{w_b\})-Q(S)}{\binom{|A^{j}|-1}{|S|}}$$
$$= \frac{1}{|A^{j}-1|}\sum_{S\subseteq A^{j}\setminus\{w_a,w_b\}}\frac{Q(S\cup\{w_a\})-Q(S\cup\{w_b\})}{\binom{|A^{j}|-2}{|S|}} \tag{12}$$

We assume that in task t_j, the worker with the lowest PSV is w_d, and the worker with the highest PSV is w_c. Hence, based on the PSVs of workers w_a and w_b, we can calculate their NPSVs more efficiently. For example, worker w_b's NPSV can be calculated as:

$$\mathbb{N}\left(\omega_b^{j}\right) = \frac{\omega_b^{j} - min(\omega^{j})}{max(\omega^{j}) - min(\omega^{j})}$$
$$= \frac{\left[\omega_a^{j} - min(\omega^{j})\right] - \left[\omega_a^{j} - \omega_b^{j}\right]}{\left[max(\omega^{j}) - \omega_a^{j}\right] - \left[min(\omega^{j}) - \omega_a^{j}\right]}$$
$$= \frac{\Delta\omega_{a,d}^{j} - \Delta\omega_{a,b}^{j}}{\Delta\omega_{c,a}^{j} - \Delta\omega_{d,a}^{j}} \tag{13}$$

Definition 12 (Approximate Shapley Value) However, due to interference from randomness, evaluating workers solely based on the NPSVs from a single task is impractical. To evaluate the workers' contributions within the team more accurately, we calculate their ASVs based on the NPSVs across all historical tasks. For worker w_i, after the completion of task t_j, his or her ASV is determined as:

$$\widehat{\theta}_i^{j} = \begin{cases} N\left(\omega_i^{j}\right), w_i \in A^{j}, p_i^{j} = 1 \\ \dfrac{b\left(p_i^{j}-1\right)}{p_i^{j}}\cdot\widehat{\theta}_i^{j-1} + \dfrac{(1-b)p_i^{j}+b}{p_i^{j}}\cdot N\left(\omega_i^{j}\right), w_i \in A^{j}, p_i^{j} \neq 1 \\ \widehat{\theta}_i^{j-1}, w_i \notin A^{j} \end{cases} \tag{14}$$

The hyperparameter $b(0 \leq b \leq 1)$ controls the update rate of a worker's ASV. In real scenarios, as tasks progress, variations of a worker's attributes, such as expertise and equipment precision, may potentially alter their Real Contribution Degrees (RCD) within the team. Therefore, it becomes necessary for the platform to dynamically adjust the relevance of historical task results based on the actual situation. For instance, setting b to 0 implies that a worker's ASV relies entirely on the NPSV obtained in the current task, disregarding all historical records. Conversely, setting it to 1 indicates the platform will completely cease the update of a worker's ASV. In this case, the platform will evaluate him or her solely based on historical records.

From Eq. (14), each worker's ASV is updated after the completion of each task. Thus, the platform can estimate the workers' contributions to the team based on their performance in all participated tasks. Specifically, when workers engage in a task for the first time, the platform considers their NPSVs from that task as their ASVs for subsequent calculations. For other workers with historical participation in tasks, their ASVs are computed as a linear combination of their ASVs from the previous tasks and their NPSVs from the current task. If workers do not participate in the current task, their ASVs remain unchanged.

Overall, since NPSV is defined as a worker's relative contribution within a team for a specific task, it exhibits significant instability. However, ASV can compensate for the limitations of NPSV and consider all historical records of tasks. This not only improves the accuracy of the platform's evaluation of a worker's contribution but also considers situations where a worker's ability may change over time, making it suitable for more complex scenarios.

Definition 13 (Weighted Sensing Quality) Similar to the format of ASV, to evaluate workers, the platform will incorporate a worker's sensing quality across historical tasks and the current task as a worker's performance might noticeably change over time. We define the weighted sensing quality of worker w_i after the completion of task t_j as:

$$\widehat{q}_i^{j} = \begin{cases} q_i^{j}, w_i \in A^{j}, p_i^{j} = 1 \\ \dfrac{b\left(p_i^{j}-1\right)}{p_i^{j}}\cdot\widehat{q}_i^{j-1} + \dfrac{(1-b)p_i^{j}+b}{p_i^{j}}\cdot q_i^{j}, w_i \in A^{j}, p_i^{j} \neq 1 \\ \widehat{q}_i^{j-1}, w_i \notin A^{j} \end{cases} \tag{15}$$

A high-confidence ASV can reasonably evaluate a worker's contribution to the team and effectively optimize the platform's decisions. However, in practical scenarios, a worker's Shapley value within the team may exhibit an unknown distribution and can be influenced by various nonlinear factors. This might result in significant fluctuations in NPSV for certain tasks. Thus, it may cause the platform's computation of a worker's comprehensive sensing ability to be less precise. In severe cases, it might lead to a collapse of the entire evaluation system. In other words, when a worker has participated in relatively few tasks, the confidence in his or her ASV will be lower. In these cases, the platform cannot evaluate the worker by directly assessing his or her ASV but needs to accumulate sufficient historical task records to improve confidence in ASV.

Specifically, when the confidence in ASVs is low, the platform leans towards evaluating workers based on their weighted sensing qualities. As workers complete more tasks, the confidence in their ASVs increases. At this point, the platform can rely more on ASVs for worker evaluation. Therefore, under different circumstances, it is crucial to dynamically integrate workers' weighted sensing qualities and their ASVs to evaluate their comprehensive sensing abilities, facilitating the platform's rational selection of workers. After the completion of task t_j, the comprehensive sensing ability of worker w_i is represented as:

$$\psi_i^{j} = \begin{cases} z\gamma_i^{j}\cdot\widehat{\theta}_i^{j} + \left(1 - z\gamma_i^{j}\right)\cdot\widehat{q}_i^{j}, w_i \in A^{j} \\ \psi_i^{j-1}, w_i \notin A^{j} \end{cases} \tag{16}$$

Here, $z(0 \leq z \leq 1)$ is a hyperparameter regulating the maximum proportion of ASV within a worker's comprehensive sensing ability. The platform can adjust z based on actual circumstances. γ_i^{j} represents the regulatory factor of worker w_i and determines the specific evaluation approach. The platform controls the proportion of his or her weighted sensing ability in the comprehensive evaluation using γ and combines it with his or her ASV to derive the current comprehensive sensing ability ψ_i^{j}. As the confidence in ASVs increases, the platform tends to rely more on ASVs for worker evaluation. The regulatory factor for worker w_i can be calculated as follows:

$$\gamma_i^{j} = \begin{cases} \dfrac{1}{1+e^{-\left(\alpha_i^{j}-\rho\right)}}, w_i \in A^{j} \\ \gamma_i^{j-1}, w_i \notin A^{j} \end{cases} \tag{17}$$

$$\alpha_i^{j} = p_i^{j}\cdot e^{-Var(\mathbb{N}(\omega_i))} \tag{18}$$

Here, ρ is a hyperparameter used to control the initialization process of the regulatory factor γ. A larger ρ requires a longer initialization process for γ. In other words, more task results are needed to increase γ significantly. From Eqs. (17–18), it is evident that γ is also influenced by the variance of a worker's historical NPSVs. This implies that if a worker's performance within the team remains consistently stable, the regulatory factor will increase more rapidly, allowing the platform to rely more on the ASV for evaluation.

It is important to note that the regulatory factor is designed in the form of a sigmoid function. Its initial growth rate is relatively low, aiming to improve confidence in ASVs through multiple tasks when the number of tasks is limited. As the reliability of ASVs gradually gets validated, the primary evaluation criterion for workers transitions from their weighted sensing qualities toward ASVs. Through this design, we have developed a rational and effective approach called IICAE. With this approach, the platform ensures the overall data quality while continually identifying higher-quality workers.

4.1.2. Detailed algorithm

According to the provided solution, IICAE is depicted in Algorithm 1. The IICAE algorithm mainly contains two layers of loops and some linear operations. Therefore, its time complexity is approximately $O(n^2)$. Initially, the platform collects the complete dataset d^j of task t_j and computes the clustered data as GTD. Following Eq. (8), the platform sequentially computes the sensing qualities of all workers involved in this task. For data involving multiple modalities, the platform uses standard Euclidean distance for computation. Specifically, the platform first calculates the standard deviation s_j of each modality. It then uses the data of each dimension collected by the worker to obtain the worker's sensing quality q_i^j in this task (Step 3).

Next, based on Eq. (10), the platform computes the PSV of each worker by evaluating the marginal contribution exhibited by each set S (Step 5). Subsequently, the platform updates the profiles of all workers involved in this task using NPSVs. According to Eq. (11), the platform applies min–max normalization to the PSVs to derive the workers' NPSVs (Step 9). According to Eq. (14), for workers without historical task records, the platform computes their ASVs, regulatory factors, and weighted sensing qualities (Steps 10–13).

For other workers participating in task t_j, the platform calculates each worker's current ASV using Eq. (14) (Step 15). Then, it computes each worker's weighted sensing quality via Eq. (15) (Step 16). Simultaneously, based on the variance of each worker's NPSV from individual historical tasks, the platform updates the worker's γ using Eq. (17) (Steps 17–18).

Finally, the platform combines the worker's ASV and weighted sensing quality to calculate the comprehensive sensing ability using Eq. (16) (Step 19). Additionally, for workers not participating in task t_j, the platform does not modify their worker profiles (Steps 21–25).

Algorithm 1: Integration of Individual and Collaborative Abilities Evaluation (IICAE)

Input:$\psi^{j-1}, \widehat{\theta}^{j-1}, \gamma^{j-1}, A^j, d^j, p^{j-1}, \widehat{q}^{j-1}, b, \rho, z$
Output:$\psi^j, \widehat{\theta}^j, \gamma^j, p^j, \widehat{q}^j$
1: **Initialize:**$i = 1, \forall t_j \in t$;
2: **for** each w_i in A^j **do**
3: $q_i^j = 1/exp\left(\sqrt{(1/P)\sum_{p=1}^{P}\left(d_i^j(p) - \overline{d}^j(p)\right)^2/s_j(p)^2}\right)$;
4: **for** each subset S of $A^j \backslash \{w_i\}$ **do**
5: $\omega_i^j += (Q(S \cup \{w_i\}) - Q(S))/\left(|A^j|\binom{|A^j|-1}{|S|}\right)$;
6: **for** each w_i in W **do**
7: **if** $w_i \in A^j$ **then**
8: $p_i^j = p_i^{j-1} + 1$;
9: $\mathbb{N}\left(\omega_i^j\right) = \left(\omega_i^j - min(\omega^j)\right)/(max(\omega^j) - min(\omega^j))$;
10: **if** $p_i^j == 1$ **then**
11: $\widehat{\theta}_i^j = \mathbb{N}\left(\omega_i^j\right)$;
12: $\gamma_i^j = 1/(1 + exp(\rho))$;
13: $\widehat{q}_i^j = q_i^j$;14: **else**
15: $\widehat{\theta}_i^j = \left(b\left(p_i^j - 1\right)/p_i^j\right) \cdot \widehat{\theta}_i^{j-1} + \left(\left((1-b)p_i^j + b\right)/p_i^j\right) \cdot \mathbb{N}\left(\omega_i^j\right)$;
16: $\widehat{q}_i^j = \left(b\left(p_i^j - 1\right)/p_i^j\right) \cdot \widehat{q}_i^{j-1} +$
$\left(\left((1-b)p_i^j + b\right)/p_i^j\right) \cdot q_i^j$;
17: $\alpha_i^j = p_i^j \cdot exp\left(-Var\left(\mathbb{N}\left(\omega_i^j\right)\right)\right)$;
18: $\gamma_i^j = 1/\left(1 + exp\left(-\alpha_i^j + \rho\right)\right)$;
19: $\psi_i^j = z\gamma_i^j \cdot \widehat{\theta}_i^j + \left(1 - z\gamma_i^j\right) \cdot \widehat{q}_i^j$;
20: **else**

(*continued on next column*)

(*continued*)

Algorithm 1: Integration of Individual and Collaborative Abilities Evaluation (IICAE)

21: $\gamma^j = \gamma^{j-1}$;
22: $\widehat{\theta}^j = \widehat{\theta}^{j-1}$;
23: $\psi_i^j = \psi_i^{j-1}$;
24: $p_i^j = p_i^{j-1}$;
25: $\widehat{q}_i^j = \widehat{q}_i^{j-1}$;
26: return $(\psi^j, \widehat{\theta}^j, \gamma^j, p^j, \widehat{q}^j)$;

4.1.3. Theoretical analysis

Theorem 1 *IICAE ensures the effectiveness of worker evaluation.*

Proof. In IICAE, we utilize the regulatory factor γ to regulate the evaluation process dynamically. During the execution of tasks, due to the randomness and fluctuations in worker performance, there tends to be a disparity between the worker's ASV and RCD. We assume that all factors contributing to the deviation of ASV from RCD can be represented by a nonlinear function $f(p,\nu)$ initiating with a large positive value. Its output is influenced by the number of the worker's historical tasks p and other nonlinear factors ν. Therefore, the ASV computed by the platform for worker w_i can be expressed as:

$$\widehat{\theta}_i^j = \theta_i^j \cdot f\left(p_i^j, \nu\right) \tag{19}$$

To minimize the error in estimating workers' contribution degrees, we need to dynamically adjust worker evaluation strategies according to the characteristics of the nonlinear function $f(p,\nu)$. When a worker participates in fewer tasks, the confidence in the worker's ASV increases relatively slowly due to the influence of randomness. However, when the worker engages in a sufficient number of tasks, the ASV approximates the RCD. Hence, despite the influence of various nonlinear factors on $f(p,\nu)$, the value of the function tends to decrease as the number of the worker's historical tasks increases. Thus, $\frac{\partial f(p,\nu)}{\partial p} < 0$. To $\forall p_a \to 0$, $\frac{\partial f(p_a,\nu)}{\partial p} \approx 0$; To $\forall p_c \to +\infty$, $f(p_c,\nu) \approx 1$, $\frac{\partial f(p_c,\nu)}{\partial p} \approx 0$. At the same time, $\exists p_b \in p$ satisfies $\frac{\partial^2 f(p_b,\nu)}{\partial p^2} = 0$, $\frac{\partial f(p_b,\nu)}{\partial p} = max\left(\frac{\partial f(p,\nu)}{\partial p}\right)$. In other words, when $p \in (0,p_b)$ or $p \in (p_b,+\infty)$, $\frac{\partial f(p,\nu)}{\partial p} < 0$; when $p = p_b$, $\frac{\partial f(p_b,\nu)}{\partial p} = min\left(\frac{\partial f(p,\nu)}{\partial p}\right)$.

According to the characteristics of the nonlinear function $f(p,\nu)$ mentioned above, we have designed the regulatory factor γ to minimize the influence of $f(p,\nu)$ on the platform. By differentiating Eq. (17), we obtain:

$$\begin{aligned}\gamma'(\alpha) &= \frac{d}{d\alpha}\left(\frac{1}{1+e^{-(\alpha-\rho)}}\right) = -\left(\frac{1}{1+e^{-(\alpha-\rho)}}\right)^2 \cdot \frac{d}{d\alpha}\left(1+e^{-(\alpha-\rho)}\right) \\ &= -\left(\frac{1}{1+e^{-(\alpha-\rho)}}\right)^2 \cdot \left(-e^{-(\alpha-\rho)}\right) = \frac{e^{-(\alpha-\rho)}}{\left(1+e^{-(\alpha-\rho)}\right)^2}\end{aligned} \tag{20}$$

From the above analysis, when $\alpha \in (0,\rho)$ or $\alpha \in (\rho,+\infty)$, it satisfies $\frac{d\gamma(\alpha)}{d\alpha} > 0$; when $\alpha = \rho$, $\frac{d\gamma(\rho)}{d\alpha} = max\left(\frac{d\gamma(\alpha)}{d\alpha}\right)$. This indicates that the regulatory factor γ exhibits an opposite trend to the nonlinear function $f(p,\nu)$. γ increases as the confidence in the worker's ASV grows. Consequently, the platform chooses a more conservative evaluation strategy when the value of $f(p,\nu)$ is relatively high. In other words, it primarily evaluates the worker's performance based on weighted sensing quality. Conversely, when $f(p,\nu)$ is lower, the platform relies more on the worker's ASV for evaluation. With this design, the platform dynamically adjusts its evaluation strategy in different scenarios.

$\square$.

4.2. Selection of team leaders

4.2.1. Basic idea

In practical situations, the social relationships existing between workers in an MCS system may have a great impact on the collaboration between them. Therefore, in the IICA-DWS algorithm, the platform will use a leader-centered social network to enhance the active collaboration within the team, thereby improving the platform efficiency as much as possible. When the platform selects workers (including team leaders and team members) to form a team, two fundamental spatiotemporal constraints must be satisfied: the duration of the task must be covered by the worker's online period, and the spatial scope of the task must be within the worker's coverage area. Only workers who meet both these essential spatiotemporal constraints and express willingness to participate in the task will be considered as candidates for team leaders $W_e(l)$ or team members $W_e(m)$ of the task. In other words, for any worker $w_i \in \{W_e(l), W_e(m)\}$ involved in a certain task, he or she must satisfy the following two constraints:

$$\begin{cases} \left[T_{t_j}^{start}, T_{t_j}^{end}\right] \subseteq \left[T_{w_i}^{start}, T_{w_i}^{end}\right] \\ L_{t_j} \subseteq L_{w_i} \end{cases} \tag{21}$$

In the IICA-DWS algorithm, the platform starts team formation with the selection of team leaders. After receiving task requests from the requester, the platform gathers all workers who are willing to become team leaders. These workers are referred to as candidate team leaders and are represented as the set $W_e(l)$. During the selection process, the primary considerations include these candidate team leaders' social influence and comprehensive sensing abilities.

The social influence reflects both the number of social friends and the strength of social connections a worker has within the current sensing area. As in Def. 7, social influence allows for a quantitative representation of a worker's significance within the social network. Selecting a team leader with strong social influence contributes to reinforcing social connections within the team and enhancing collaboration among workers.

Furthermore, when there are k tasks with overlapping sensing areas occurring simultaneously, the platform will select the top k candidate team leaders according to the comprehensive rankings. And the platform will select a leader for each task in order. We believe that candidates with higher comprehensive rankings are more likely to organize teams with higher efficiency. Consequently, team leaders with higher comprehensive rankings will be assigned to tasks with higher budgets.

Specifically, when the platform is about to execute new tasks, it selects appropriate team leaders of the tasks in the following steps:

1) Search for all workers who meet the temporal and spatial requirements of these tasks.
2) Issue a recruitment call for team leaders and gather all candidates.
3) Rank the social influence and comprehensive sensing abilities of all candidate team leaders separately, denoted as $Rank(\zeta)$ and $Rank\left(\psi^j\right)$.
4) Calculate the comprehensive rankings of all candidate team leaders: $Rank(W) = Rank(\zeta) \times Rank\left(\psi^j\right)$.
5) Select the top-ranked k candidates as the team leaders, denoted as $W_s(l)$.

Algorithm 2: Selection of team leaders

Input: $W_e(l), \mathbb{C}, y, \psi^j, k$
Output: $W_s(l)$
1: **Initialize:** $i = 1$, $Rank(\zeta) = W_e(l), Rank\left(\psi^j\right) = W_e(l), \forall t_j \in t$;
2: **for** each w_i in $W_e(l)$ **do**
3: $\quad \zeta_i = \sum_{k=1}^{n} \mathbb{C}_{ik} \cdot y_{ik}$;
4: **for** a from 1 to $|W_e(l)| - 1$ **do**
5: $\quad$ **for** b from 1 to $|W_e(l)| - a$ **do**
6: $\quad\quad$ **if** $\zeta_b < \zeta_{b+1}$ **then**
7: $\quad\quad\quad$ Swap $Rank(\zeta_b)$ and $Rank(\zeta_{b+1})$;
8: $\quad\quad$ **if** $\psi_b^j < \psi_{b+1}^j$ **then**
9: $\quad\quad\quad$ Swap $Rank\left(\psi_b^j\right)$ and $Rank\left(\psi_{b+1}^j\right)$;
10: $Rank(W) = Rank(\zeta) \cdot Rank\left(\psi^j\right)$;
11: **for** top kw_i in $Rank(W)$ **do**
12: $\quad W_s(l) = W_s(l) + \{w_i\}$;
13: return $W_s(l)$;

4.2.2. Detailed algorithm

According to the basic idea in section 4.2.1, the IICA-DWS algorithm selects team leaders through Algorithm 2. The platform gathers all workers who are willing to become team leaders and meet the spatiotemporal constraints of the task as candidate team leaders $W_e(l)$. If there are k tasks with identical spatiotemporal constraints, the platform will select the top k candidates according to comprehensive rankings and make these workers team leaders.

Firstly, the platform utilizes Eq. (2) to consider both the total number of social connections and the social strength of these relationships, in order to compute the candidates' social influence ζ (Step 3). Subsequently, based on the social influence, the platform ranks the candidate team leaders and acquires the ranking $Rank(\zeta)$ (Step 7).

Similarly, the platform ranks the comprehensive sensing abilities ψ^j of the candidate team leaders and gets the ranking $Rank\left(\psi^j\right)$ (Step 9).

Finally, the platform multiplies these two rankings to compute the comprehensive ranking of all candidate team leaders, denoted as $Rank(W)$ (Step 10). Subsequently, based on this ranking, the platform determines the set of chosen team leaders, represented by $W_s(l)$ (Steps 11–12).

4.3. Selection of team members

4.3.1. Basic idea

After selecting the team leaders, the platform chooses appropriate team members under the constraints of Eqs. (5–6). The remaining unsuccessful candidate team leaders and other potential participants for this task become candidate team members, denoted by the set $W_e(m)$. In this section, we will outline the specific method the IICA-DWS algorithm uses to select team members.

MAB is a widely used reinforcement learning model for making online decisions in uncertain environments. Essentially, the MAB model is a bandit with multiple arms. Each arm is associated with rewards obtained from an unknown distribution. Players aim to maximize rewards by strategically exploring and exploiting the arms of the bandit within a limited number of chances.

In an MCS system, on the one hand, the platform needs to select the best workers through online evaluations under specific conditions. On the other hand, the results of each worker completing tasks follow some unknown distribution. Thus, the selection of team members can be viewed as an MAB problem. The platform's selection of workers is identical to the action of pulling the arms of bandits in essence. The results of task completion can be represented as the rewards of bandits. In MAB problems, players aim to maximize their cumulative rewards. Similarly, the platform aims to recruit superior workers at lower cost and achieve higher platform efficiency in an MCS system.

In the member selection process, the platform should consider two factors: first, gaining more knowledge about newly added workers and those with lower confidence in their performance (so-called exploration) to identify better workers; second, selecting workers with higher confidence in performance and better cost-effectiveness to join the team (so-called exploitation). The selection problem of team members can be addressed using the MAB model, which essentially represents an online learning and decision-making process.

In the initial phase, the platform recruits workers tentatively to understand their efficiency in task execution. Since the performance of these workers is unknown at this stage, each worker is treated equally. The platform examines these workers sequentially, recruiting the first set of workers $\{w_1, \cdots, w_i\}$, followed by the second set $\{w_{i+1}, \cdots, w_j\}$, and so forth, ensuring that all workers have completed at least one task.

After the release of task t_j, the platform conducts a reverse auction where candidate workers report their bids $\beta^j = \left\{\beta^j_1, \beta^j_2, \cdots, \beta^j_n\right\}$ for the current task. Simultaneously, the platform calculates the maximum bid for each worker as $\beta^j_{max} = B_j/R_j$ and ensures that the number of workers employed by the platform is not less than R_j and the total cost does not exceed the total budget B_j. If a candidate worker's reported bid exceeds β^j_{max}, the platform disregards his or her task request directly.

In order to complete the tasks successfully, all candidates need to meet the spatiotemporal constraints of tasks demonstrated in Eq. (21). Additionally, the platform must select workers with high comprehensive sensing abilities while paying the lowest possible bids. If a worker possesses strong ability but quotes a high bid, he or she might consequently reduce platform efficiency. Thus, solely relying on a worker's comprehensive sensing ability for evaluation is not enough; his or her bid should also be taken into account. We define the Revenue-Cost-Ratio (RCR) based on the comprehensive sensing ability and bid as $h^j_i = \psi^j_i/\beta^j_i$. Through workers' RCRs, the platform can precisely evaluate the efficiency of each worker in completing tasks. Therefore, it can develop more efficient recruitment strategies.

According to Def. 5, stronger social connections among team members and team leaders can foster a tighter social bond within the team, thereby enhancing collaborative behaviors. Thus, in the process of selecting team members, it is essential to evaluate the social connections between candidate team members and the team leader.

Additionally, in typical scenarios, the workers' comprehensive sensing abilities follow an unknown distribution. This causes uncertainty in the short term. Thus, it may potentially lead to inaccurate evaluations of workers. In real scenarios, directly using workers' RCRs for evaluation may make it almost impossible to re-recruit certain highly efficient members due to their occasional low-efficiency performances. Hence, we define the Upper Confidence Bound (UCB) index to address this issue, represented by the set $h^{j+} = \left\{h^{j+}_1, h^{j+}_2, \cdots, h^{j+}_n\right\}$. After the completion of task t_j, the UCB index of worker w_i is defined as:

$$h^{j+}_i = h^j_i + \kappa \mathbb{C}_{il} \cdot y_{il} + \xi^j_i \tag{22}$$

$$\xi^j_i = \sqrt{\frac{\delta \cdot ln\left(\sum_{k=1}^{n} p^{j-1}_k\right)}{p^{j-1}_i}} \tag{23}$$

Here, $\kappa (0 \leq \kappa \leq 1)$ is a hyperparameter. Depending on various scenarios in the real world, the platform can adjust κ to modify the weight of the social intensity between the candidate team members and the team leader during the selection process. ξ^j_i serves as an additive factor. It gives less frequently chosen workers more chances to be recruited again. δ, as a hyperparameter, offers flexibility to our strategy. The platform must adjust its value according to the actual circumstances in order to approximate the optimal results. This design helps reduce the impact of randomness on the platform. Thus, the platform is enabled to explore the workers more thoroughly.

During the team formation process for task t_j, the platform continuously selects workers with the highest UCB indexes to join the team until the budget B_j for that task is exhausted. After the task is completed, the platform records and updates all the relevant attributes of the workers (including ASV, UCB index, etc.) and awaits the next task.

Algorithm 3: Selection of team members

Input: $W_e(m), \beta^j, \psi^j, p^{j-1}, B_j, R_j, \mathbb{C}_l, y_l, \kappa, \delta$
Output: $W_s(m)$
1: **Initialize**: $i = 1$, $Rank\left(h^{j+}\right) = W_e(m)$, $\forall t_j \in t$;
2: $\beta^j_{max} = B_j/R_j$;
3: **for** each w_i in $W_e(m)$ **do**

(continued on next column)

(continued)

Algorithm 3: Selection of team members

4: **if** $\beta^j_i > \beta^j_{max}$ **then**
5: $W_e(m) = W_e(m) - \{w_i\}$;
6: **if** $p^{j-1}_i == 0$ **then**
7: $W_s(m) = W_s(m) + \{w_i\}$;
8: $W_e(m) = W_e(m) - \{w_i\}$;
9: $h^j_i = \psi^j_i/\beta^j_i$;
10: $\xi^j_i = \sqrt{\delta \cdot ln\left(\sum_{k=1}^{n} p^{j-1}_k\right)/p^{j-1}_i}$;
11: $h^{j+}_i = h^j_i + \kappa \mathbb{C}_{il} \cdot y_{il} + \xi^j_i$;
12: **for** a from 1 to $|W_e(m)| - 1$ **do**
13: **for** b from 1 to $|W_e(m)| - a$ **do**
14: **if** $h^{j+}_b < h^{j+}_{b+1}$ **then**
15: Swap $Rank\left(h^{j+}_b\right)$ and $Rank\left(h^{j+}_{b+1}\right)$;
16: **for** w_i in $Rank\left(h^{j+}\right)$ **do**
17: **if** $B_j \geq \beta^j_i$ **then**
18: $W_s(m) = W_s(m) + \{w_i\}$;
19: $B_j = B_j - \beta^j_i$;
20: **else**
21: **break** ;
22: return $W_s(m)$;

4.3.2. Detailed algorithm

Based on the analysis above, the IICA-DWS algorithm leverages the MAB model for selecting team members, as outlined in Algorithm 3. Initially, the platform determines the maximum bid β^j_{max} for an individual worker based on the requirements of task t_j. Specifically, the requirements include the minimum team size R_j and the budget B_j determined in advance (Step 2). If the bid of any candidate team member exceeds β^j_{max}, the platform will not select him or her for the task since the platform considers the overall efficiency and data quality (Steps 4–5). To efficiently evaluate the comprehensive sensing abilities of new workers, the platform gives them priority to participate (Steps 6–8).

The platform calculates the candidate team members' RCRs based on their comprehensive sensing abilities ψ^j_i and the bids β^j_i they reported for the current task (Step 9). Additionally, the platform computes the social intensity $\mathbb{C}_{il}$ between each candidate team member and the team leader. Simultaneously, using Eq. (23), the platform calculates the additive factor ξ^j_i based on the number of tasks completed by the candidate team members (Step 10). Using these three parameters, the platform computes the UCB indexes h^{j+}_i for all remaining candidate team members (Step 11) and ranks them to obtain $Rank\left(h^{j+}\right)$ (Steps 12–15).

Finally, the platform recruits workers continuously based on $Rank\left(h^{j+}\right)$ until the budget B_j is exhausted. At that point, the platform will have the set of workers participating in this task, denoted as $W_s(m)$ (Steps 16–21).

4.3.3. Theoretical analysis

In this section, we will demonstrate the rationality and fairness of the platform through the IICA-DWS algorithm in recruiting workers and forming teams.

Theorem 2 *IICA-DWS algorithm satisfies the principle of platform rationality.*

Proof In the IICA-DWS algorithm, the platform determines the total budget B_j at the beginning of each task, which also serves as the maximum bid for the team. This ensures that the platform does not suffer from losses. The platform calculates the maximum task cost for each worker participating in the task, denoted as $\beta^j_{max} = B_j/R_j$. This guarantees that the platform can reliably make profits while ensuring task completion quality. Additionally, the platform can dynamically adjust B_j based on its actual situation as long as the total budget does not exceed the total payment from the requester. Therefore, the IICA-DWS algorithm complies with the principle of platform rationality.

□.

Theorem 3 *IICA-DWS algorithm satisfies the principle of fairness.*

Proof To better illustrate fairness, we may consider a simplified MCS system based on the IICA-DWS algorithm, where this system consistently selects k workers as a set for each task. To illustrate more clearly, we assume that in this system, the impact of the additive factor is relatively minimal. For two workers, w_a and w_b, assume they both have relatively high UCB indexes and have participated in multiple tasks while having no social connections with the team leader, and $\psi_a^j < \psi_b^j$. For worker w_a, his or her RCR is denoted as $h_a^j = \psi_a^j / \beta_a^j$. We may assume that w_a ranks as the $(k-1)$-th worker according to RCR, while the k-th worker's RCR is denoted as h'. In subsequent tasks, workers w_a and w_b need to ensure that their RCRs are greater than h' to guarantee more opportunities for participation. In a critical scenario where $h_a^{j+1} = h_b^{j+1} = h'$, it follows that $\beta_a^{j+1} = \psi_a^{j+1}/h'$ and $\beta_b^{j+1} = \psi_b^{j+1}/h'$, which results in $\beta_a^{j+1} < \beta_b^{j+1}$. This indicates that compared to worker w_a, worker w_b can appropriately increase his or her reported bid while ensuring continued selection by the platform, thereby attaining higher profits.

However, in a real MCS system, the team's size might vary, and a worker's sensing result in a particular task could be influenced by incidental factors related to themselves or external conditions. Hence, worker w_b has to decide whether to maintain or decrease the bid to ensure continued selection by the platform, as he or she aims for a more stable long-term benefit. From the analysis above, it is evident that elevated RCRs lead to more profits and opportunities for the workers to participate in tasks. Therefore, the IICA-DWS algorithm satisfies the principle of fairness.

□.

5. Performance evaluation

5.1. Simulation setup

We conducted numerous simulation experiments using the Gowalla dataset. The dataset includes both friend relationships and check-in records. In Fig. 4, we consider users within the Gowalla dataset as workers, while their check-in records serve as tasks for our simulations. Specifically, the Gowalla dataset comprises information about social connections among various users, which we use to simulate the social network of workers within the MCS system. Additionally, the dataset's check-in records contain time and location information, which we interpret as temporal and spatial constraints of tasks. The simulation settings are shown in Table 2.

In our simulation experiments, we use a Gaussian distribution to represent the individual sensing quality $\check{q}_a$ of a single worker w_a when not collaborating with others, introducing a certain level of randomness. That is, $\check{q}_a \sim \mathcal{N}(\mu_a, {\sigma_a}^2)$. We assume that workers with stronger social influence might engage in more collaborative behaviors with others. According to Def. 4, we define $\chi = \{\chi_1, \chi_2, \cdots, \chi_n\}$ to represent the collaborative abilities of individual workers. The data collected by workers in a particular task is influenced by both their individual and collaborative abilities.

Table 2
Simulation settings.

Parameter name	Values
Number of tasks	[50, 550]
Number of workers	[50, 150]
b	[0.1, 0.9]
z	[0.2, 1.0]
δ	[0, 0.8]
ρ	[10, 30]
CVISQ $c_v(\check{q})$	[0.5, 1.0]
CVCA $c_v(\chi)$	[0.5, 1.5]

Based on these assumptions, $\check{q}_a$ reflects the ability of worker w_a to independently perform tasks, while χ_{ab} denotes the collaborative degree between w_a and w_b. Therefore, for a worker w_a within a team of n members, the sensing quality during task t_j can be mathematically represented as:

$$q_a^j = \check{q}_a^j \cdot \sum\nolimits_{b=1, a \neq b}^{n} \chi_{ab} \tag{24}$$

Next, we explored the impact of variations in workers' individual sensing qualities and collaborative abilities on the platform. Due to dimensional uncertainties, when conducting simulation experiments, we use the coefficients of variation as a reference to provide a clearer illustration of the impact of collaboration on the team. In the simulation experiments, we adjusted the coefficient of variation for the individual sensing qualities $c_v(\check{q})$ (CVISQ) and the coefficient of variation for collaborative abilities $c_v(\chi)$ (CVCA) of all workers to simulate MCS systems with different levels of collaboration, defined as follows:

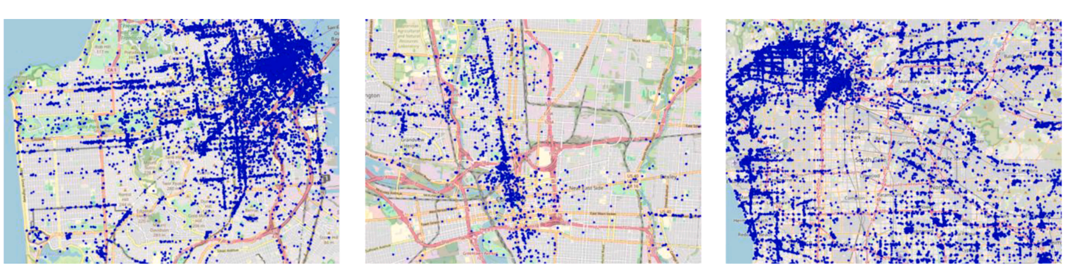
$$c_v(\chi) = \frac{m \cdot n \cdot Var(\chi)}{\sum_{j=1}^{m} \sum_{i=1}^{n} \chi_i^j} \tag{25}$$

$$c_v(\check{q}) = \frac{m \cdot n \cdot Var(\check{q})}{\sum_{j=1}^{m} \sum_{i=1}^{n} \check{q}_i^j} \tag{26}$$

5.2. Algorithms for comparison

Given that our IICA-DWS algorithm primarily focuses on worker evaluation and team formation in MCS systems with collaboration, there are no existing algorithms directly comparable to ours. Hence, we selected algorithms that closely approximate this problem. Specifically, we chose IICA-WS, ε-Greedy, and Random as comparative algorithms.

In the Random algorithm, the platform randomly selects workers whose reported bids do not exceed $\beta_{\max}$ to participate in the tasks. The IICA-WS algorithm is a variant of the IICA-DWS algorithm. Unlike the IICA-DWS algorithm, the IICA-WS algorithm does not adjust the worker evaluation system dynamically. In other words, the regulatory factor γ remains as a constant. In our simulation experiments, we set $\gamma = 0.5$ and $z = 1$ for the IICA-WS algorithm. It directly reflects the effectiveness of the dynamic evaluating method in the IICA-DWS algorithm through γ,



(a) (b) (c)

Fig. 4. Overview of the dataset.

compared with the IICA-WS algorithm.

Unlike both the IICA-DWS and IICA-WS algorithms, the ε-Greedy algorithm completely disregards collaboration within the team. It calculates the RCRs based on workers' historical average sensing qualities $\overline{q}$ and the current bids β, i.e., $h_i^{j+} = \overline{q}_i^j/\beta_i^j$, and uses RCR as an indicator for greedily selecting workers. The ε-Greedy algorithm employs $\varepsilon(0 \le \varepsilon \le 1)$ to determine the specific worker selection strategy, where ε determines the proportion used for exploration and exploitation by the platform. While ensuring platform efficiency, the platform aims to gain comprehensive knowledge of candidate workers. Specifically, the platform randomly explores workers with a probability of ε, and it exploits by greedily selecting workers with the highest RCRs to join the team with a probability of $(1-\varepsilon)$. We chose the ε-Greedy algorithm with $\varepsilon = 0.3$ as a comparative algorithm in our simulation experiments.

Azzam (Azzam, 2016) proposed the Group-Based Recruitment System (GRS), which aims to effectively recruit participants to complete tasks. GRS selects participants based on a genetic algorithm and evaluates the quality and confidence level of the entire group. Instead of evaluating the quality of each participant individually, GRS considers the overall performance of the entire group. The system forms groups based on the distribution of participants in the sensing area, their equipment capabilities, and historical performance, and calculates the score of each group. Their experiments show that GRS can effectively manage resources, optimize the recruitment process of participants, and improve the overall efficiency and completion quality of MCS tasks. In addition, Stable-GRS (Azzam, Mizouni, et al., 2018) is a version optimized for platform stability based on GRS. It has good effects in improving fairness, reducing computation time, and improving quality of information. After using GRS as the benchmark in the experiment, we found that the performance improvement is similar to that of Stable-GRS. Stable-GRS is an optimization method that maintains the stability of continuous sensing group recruitment, while the goal of the IICA-DWS algorithm is to use the collaboration within the teams to improve platform efficiency. It is unfair to compare them directly. Therefore, in our experiments, GRS was selected as an algorithm for comparison.

5.3. Evaluation results

In this section, we will analyze the performance of various algorithms under different experimental conditions. Experimental results indicate that the IICA-DWS algorithm notably outperforms the comparative algorithms in terms of platform efficiency and cumulative regret in all scenarios. Furthermore, we conducted more specific experiments to demonstrate that the IICA-DWS algorithm achieves the highest data quality under the same budget.

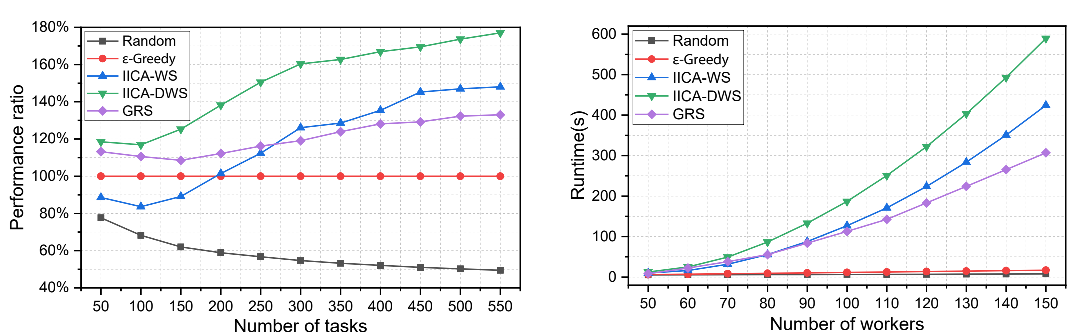
Subsequently, we delved deeper into experiments about worker identification and selection within the IICA-DWS algorithm, showcasing its exceptional ability to identify and select excellent workers to join the team. Moreover, we observed that the IICA-DWS algorithm exhibits remarkable stability in worker identification across different conditions.

We conducted individual experiments on the hyperparameters within the IICA-DWS algorithm to explain their functions. Then, we suggested how the platform should adjust them based on actual circumstances.

Firstly, we explored the different impacts of CVISQ and CVCA within the team on the platform efficiency and cumulative regret of the IICA-DWS algorithm and comparative algorithms. Here, platform efficiency is defined through Eq. (4). Regret indicates the difference between the algorithm's decision and the optimal solution given the current situation. From the experimental results, it is evident that the IICA-DWS algorithm outperforms all other algorithms in terms of platform efficiency and cumulative regret across various scenarios. Specifically, the performance ratio of IICA-DWS, IICA-WS, ε-Greedy, and Random is depicted in Fig. 5(a). It shows that IICA-DWS performed at least a 16.8 % improvement compared to ε-Greedy. Furthermore, the performance of IICA-DWS even exceeded three times compared to that of Random. As for GRS, it performs relatively stably in various situations. When the number of tasks is small, it performs significantly better than IICA-WS, and even close to the performance of IICA-DWS. Compared with GRS, IICA-DWS has an average performance improvement of 25 %, and a maximum performance improvement of 33 %. As shown in Fig. 5(b), except for Random, the runtime of other algorithms increases significantly with the increase of the number of workers. Among all algorithms, IICA-DWS always has the longest runtime. When the number of workers is less than 80, the runtime of IICA-WS is often less than that of GRS. When the number of workers is larger, IICA-WS becomes the second longest running algorithm, and its growth rate is approximately equal to that of IICA-DWS. Specifically, when the number of workers is 100 and the number of tasks is 300, compared to GRS, IICA-WS takes 112.4 % of the time, while IICA-DWS takes 165.5 % of the time.

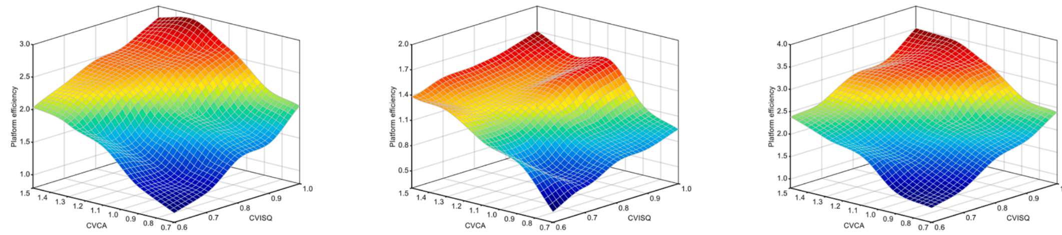
From Fig. 6, it is evident that as the number of tasks increases, IICA-DWS can compute the contributions of workers within the team more accurately. Thus, it enables more outstanding workers to join teams, thereby enhancing platform efficiency. Additionally, the platform efficiency achievable by IICA-DWS demonstrates a monotonic increase with the rise in CVCA and CVISQ, as shown in Fig. 6. This is because an increase in CVCA or CVISQ implies a greater number of workers with strong individual or collaborative abilities available within the system. With more such workers, IICA-DWS has more opportunities to explore and exploit outstanding workers. Consequently, the IICA-DWS algorithm can significantly improve platform efficiency.

Specifically, compared with IICA-WS and ε-Greedy, IICA-DWS showed an average improvement of 27.1 % and 50.8 % in platform efficiency, reducing regret by an average of 22.1 % and 27.3 %, respectively. As seen in Figs. 7 and 9, when the workers' CVCA was higher, such as $c_v(\chi) = 1.5, c_v(\check{q}) = 0.8$, IICA-DWS showed a 62.1 % improvement in platform efficiency over the ε-Greedy algorithm on average. However, when the workers' CVISQ was higher, with $c_v(\chi) = 1.1$, $c_v(\check{q}) = 1.0$, IICA-DWS only achieved an average increase in platform efficiency of 25.8 % over ε-Greedy. This is because individual workers'



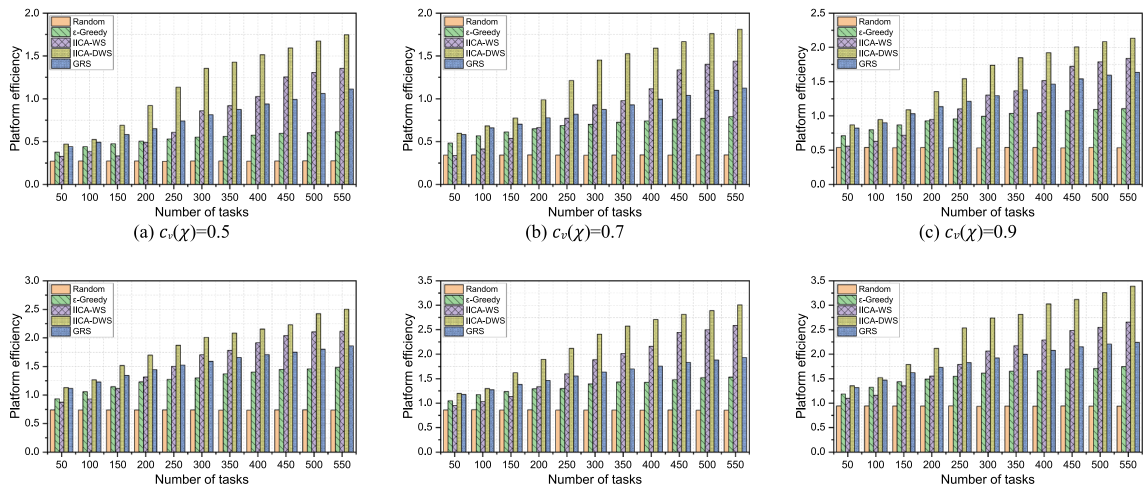
(a) (b)

Fig. 5. Performance and runtime comparison.



(a) M=100 (b) M=300 (c) M=500

Fig. 6. Platform efficiency vs. Number of tasks M.



(a) $c_v(\chi)$=0.5 (b) $c_v(\chi)$=0.7 (c) $c_v(\chi)$=0.9

(d) $c_v(\chi)$=1.1 (e) $c_v(\chi)$=1.3 (f) $c_v(\chi)$=1.5

Fig. 7. Platform efficiency vs. CVCA $c(\chi)$.

sensing qualities and collaborative abilities can be considered as two opposing factors in the MCS system. Hence, when one factor's coefficient of variation is larger, the effect of the other factor on platform efficiency is relatively weakened. It is worth noting that IICA-DWS usually performs roughly the same as GRS in the early stages, but significantly outperforms GRS in the later stages. This is because IICA-DWS needs to explore workers more when there are fewer tasks. This severely limits the growth of its platform efficiency.

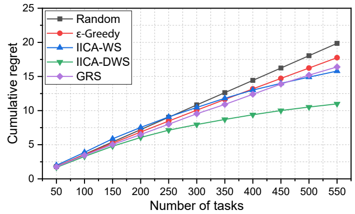
In other words, when CVISQ is higher, the individual sensing qualities have a more significant impact on platform efficiency. It means that the effect of collaboration on the team is relatively smaller. Consequently, the performance gap between ε-Greedy and IICA-DWS narrows. Conversely, when CVCA is larger, the collaboration among workers has a greater effect on the sensing results. IICA-DWS can calculate this impact better. This results in relatively higher platform efficiency.

In Fig. 7, it is noticeable that when CVCA is higher, the collaboration within the MCS system is more apparent, and the ε-Greedy algorithm's performance is substantially restricted. Additionally, when the number of tasks is low, a larger CVCA leads all algorithms to perform closer to Random, displaying relatively poorer performance. For instance, as shown in Fig. 7, when $c_v(\chi) = 0.9$ and there were 50 tasks, IICA-DWS improved platform efficiency by 61.3 % over Random. However, at $c_v(\chi) = 1.5$, this improvement decreased to 43.5 %. This is because both IICA-DWS and ε-Greedy algorithms are in an exploration phase with fewer tasks. They both primarily assess workers based on sensing qualities, while IICA-WS equally employs low-confidence ASVs and sensing qualities to evaluate workers. In such a scenario, a larger CVCA weakens the impact of individual sensing qualities on results and significantly reduces the effectiveness of the algorithms. Thus, it makes the algorithms' performances approach Random.

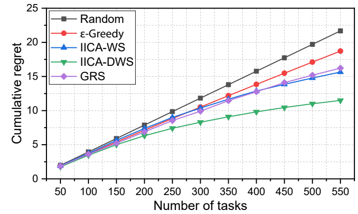
As the number of tasks increases, the IICA-DWS's calculation of workers' ASVs becomes closer to their RCDs, resulting in a better performance than ε-Greedy. Figs. 8 and 10 indicate that the growth rate of IICA-DWS's cumulative regret gradually decreases as the number of tasks increases. Eventually, the cumulative regret converges to a relatively small value. This is because when there are fewer tasks, IICA-DWS focuses on exploring workers while mainly relying on weighted sensing qualities for evaluation. Hence, in the early stages, IICA-DWS performs similarly to ε-Greedy, with similar rates of regret accumulation, as shown in Figs. 8 and 10. The difference between them is around 9.5 % on average. However, as the confidence in ASVs and regulatory factors γ increase, the platform calculates workers' ASVs more accurately. In this scenario, the calculated ASVs are much closer to the RCDs of workers. Consequently, the cumulative regret generated by IICA-DWS gradually diminishes, indicating its ability to select superior workers for the team, thereby reducing the platform's cumulative regret.

Overall, although the IICA-WS algorithm utilizes ASVs for the comprehensive evaluation of workers, the ratio between ASV and weighted sensing quality is fixed at 1. Hence, in scenarios with fewer tasks, the judgments made about the workers are inaccurate due to the low confidence in ASVs. Consequently, the IICA-WS algorithm might exhibit poorer performance than the ε-Greedy algorithm in this case. Specifically, as shown in Figs. 7 and 9, when there are 100 tasks, the platform efficiency of the IICA-WS algorithm was about 12.3 % lower than that of the ε-Greedy algorithm on average.

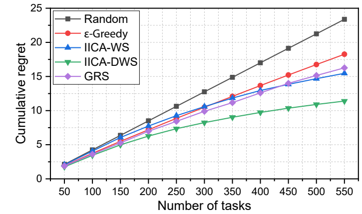
However, as the number of tasks increases and the platform gains higher confidence in workers' ASVs, the design similar to IICA-DWS allows the IICA-WS algorithm to better leverage collaboration among workers for evaluation, thereby showing better performance than the ε-Greedy algorithm. According to Figs. 7 and 9, when the number of tasks reached 200, the platform efficiency of the IICA-WS surpassed that of the ε-Greedy by around 7.1 % and continued to increase with more



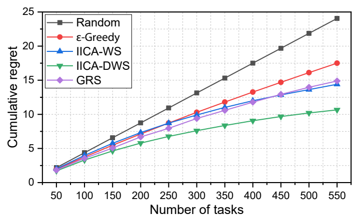
(a) $c_v(\chi)$=0.5



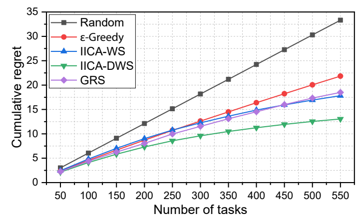
(b) $c_v(\chi)$=0.7



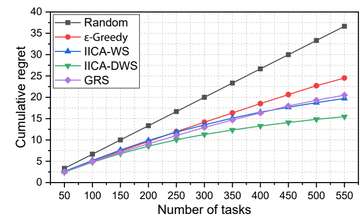
(c) $c_v(\chi)$=0.9



(d) $c_v(\chi)$=1.1

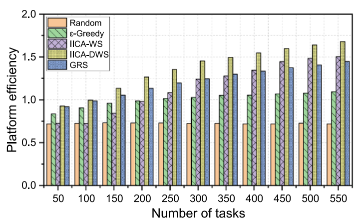


(e) $c_v(\chi)$=1.3

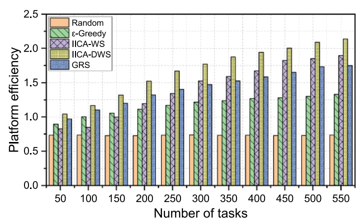


(f) $c_v(\chi)$=1.5

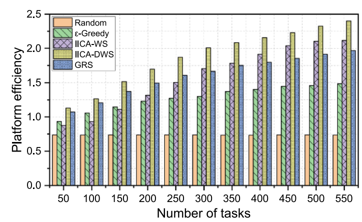
Fig. 8. Cumulative regret vs. CVCA $c(\chi)$.



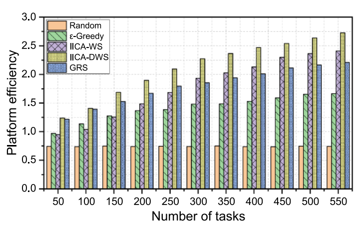
(a) $c_v(\breve{q})$=0.5



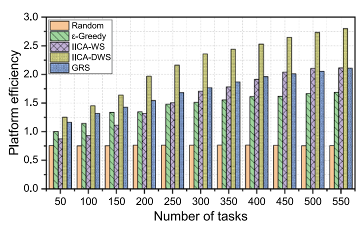
(b) $c_v(\breve{q})$=0.6



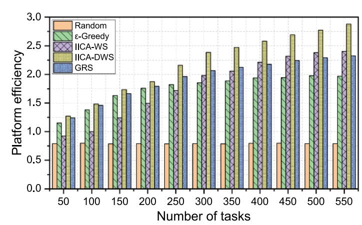
(c) $c_v(\breve{q})$=0.7



(d) $c_v(\breve{q})$=0.8

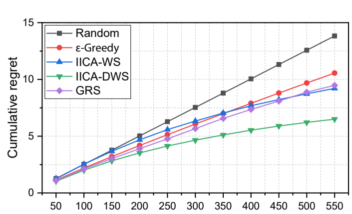


(e) $c_v(\breve{q})$=0.9

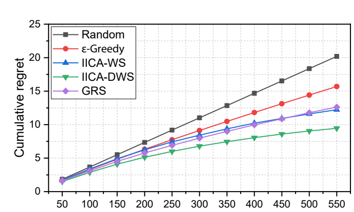


(f) $c_v(\breve{q})$=1.0

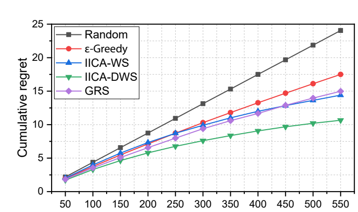
Fig. 9. Platform efficiency vs. CVISQ $c(\breve{q})$.



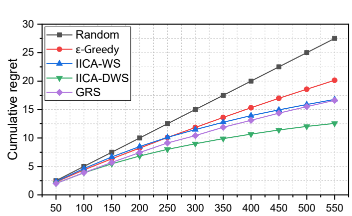
(a) $c_v(\breve{q})$=0.5



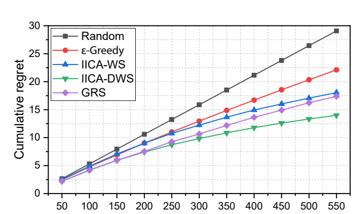
(b) $c_v(\breve{q})$=0.6



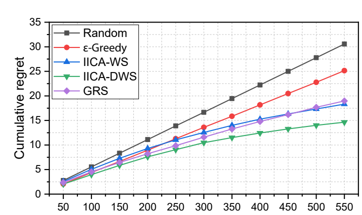
(c) $c_v(\breve{q})$=0.7



(d) $c_v(\breve{q})$=0.8



(e) $c_v(\breve{q})$=0.9



(f) $c_v(\breve{q})$=1.0

Fig. 10. Cumulative regret vs. CVISQ $c(\breve{q})$.

tasks. Particularly, with fewer tasks, the platform efficiency of IICA-WS was approximately 26.3 % lower than that of IICA-DWS. Although its performance improves noticeably with more tasks, it cannot fully utilize the higher-confidence ASVs due to its fixed regulatory factors. Consequently, it cannot ultimately achieve performance similar to IICA-DWS. In the best scenario, the platform efficiency of the IICA-WS algorithm was still approximately 13.1 % lower than that of IICA-DWS. Due to the characteristics of GRS, it can have a more stable performance in the whole process compared to IICA-WS. Specifically, this is because it does not need to conduct detailed individual evaluations of each worker, but instead scores groups, which greatly reduces the cost of the platform spent on exploration, but also makes it unable to have the same high performance as IICA-DWS. According to Figs. 7–10, under all experimental conditions, IICA-DWS has an average performance improvement of 18.2 % and a 26.8 % reduction in regret compared to GRS.

Additionally, we investigated the impact of the total budget B on the platform's data quality. We conducted several experiments within the range of [100, 150]. And B was increased by 10 in each experiment.

As depicted in Fig. 11, the experiments demonstrate that the IICA-DWS algorithm not only guarantees the highest platform efficiency but also maintains the highest data quality under the same budgets. According to Fig. 11, the data quality of the IICA-DWS algorithm was higher on average by 19.7 % and 47.3 % compared to the IICA-WS and ε-Greedy algorithms under the same budget, respectively. It is worth noting that at lower values of B, both the IICA-DWS and IICA-WS algorithms exhibited slower growth rates in data quality. Conversely, at higher values of B, these algorithms showed a more pronounced increase. Specifically, at $B = 100$ and $B = 150$, the average growth rate in data quality for the IICA-DWS and IICA-WS algorithms increased by approximately 77 %. This indicates that the performance of these two algorithms was limited at $B = 100$ due to the lower budget constraint. When the budget is constrained, the IICA-DWS and IICS-WS algorithms allocate a larger proportion of the budget for worker exploration. Thus, the proportion of the budget utilized for exploitation is reduced, thereby having a significant negative impact on the overall data quality. Overall, compared with IICA-WS, GRS can provide reliable and stable data quality under limited budgets. It is worth noting that under limited budgets, GRS performs significantly better than IICA-WS when the number of tasks is less than 250, but performs slightly worse than IICA-WS when there are more tasks. Compared with GRS, IICA-DWS has an average data quality improvement of 19.1 % under all experimental conditions.

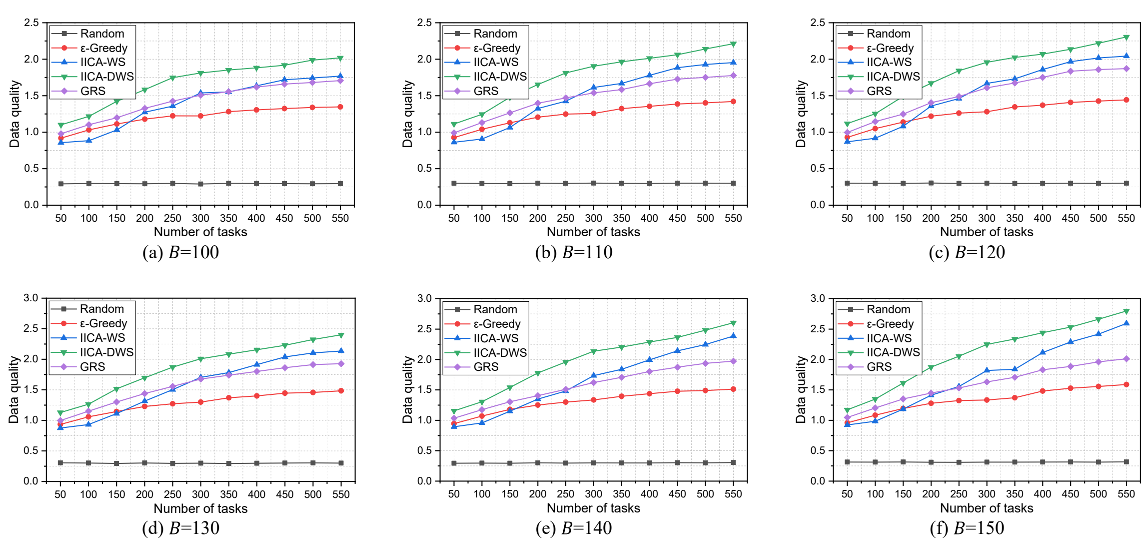
To evaluate the algorithm's worker identification ability, we compared workers' RCDs with the ASVs estimated by the IICA-DWS algorithm, as shown in Fig. 12(a). Overall, the IICA-DWS algorithm's estimation of RCD through ASV was relatively accurate, with an average discrepancy of about 5.1 %. Furthermore, we noticed that the judgment of the IICA-DWS algorithm was more precise for workers with higher RCDs. Specifically, for workers with RCDs above 0.6, the estimation error of the IICA-DWS algorithm was around 2.6 %. This is because workers with higher ASVs in the team tend to be recruited more frequently by the platform and accumulate more task records, leading to a more accurate evaluation of their ASVs by the platform. However, a small fraction of workers with strong comprehensive sensing abilities might have lower RCRs due to their higher bids. This limits the growth of their recruitment frequency. Thus, there are some obvious deviations when we apply the IICA-DWS algorithm to identify these workers.

To further illustrate the IICA-DWS algorithm's performance in worker selection, we conducted simulation experiments using a simplified MCS system. The results depicted in Fig. 12(b) indicate that the IICA-DWS algorithm thoroughly explores workers in the early stages and accurately selects workers with higher RCDs in the subsequent stages. Consequently, in scenarios with fewer tasks, the difference in performance between the IICA-DWS algorithm and the comparative algorithms was minimal. However, as the number of tasks increases, the advantages of the IICA-DWS algorithm become increasingly apparent. The algorithm effectively balances exploration and exploitation phases within a limited number of tasks.

We evaluated the identification errors of workers in teams with different CVCAs and CVISQs using the IICA-DWS algorithm. Specifically, we conducted experiments with CVCAs set to 0.5, 1.0, and 1.5 and CVISQs set to 0.5, 0.75, and 1.0. As illustrated in Fig. 13, as the number of tasks increased, the identification errors under various CVCAs and CVISQs conditions significantly decreased to low values. Especially when the number of tasks reached 550, the identification errors were reduced by 73.7 % and 72.7 %, respectively. This proves that the IICA-DWS algorithm becomes more accurate in identifying workers as the number of tasks increases. Remarkably, even with higher CVCAs or CVISQs, the identification errors of the IICA-DWS algorithm consistently converged to extremely low values. This showcased the algorithm's adaptability to different conditions and strong stability.

Finally, we investigated the impact of various hyperparameters within the IICA-DWS algorithm under different experimental conditions on platform efficiency. We comprehensively analyzed the experimental results to illustrate their respective functions within the algorithm.

The update rate of ASV and weighted sensing quality: We conducted multiple experiments within the range of [0.1, 0.9] for the hyperparameter b and increased it by 0.2 in each trial. The results are



(a) B=100 (b) B=110 (c) B=120 (d) B=130 (e) B=140 (f) B=150

Fig. 11. Data quality vs. Budget B.

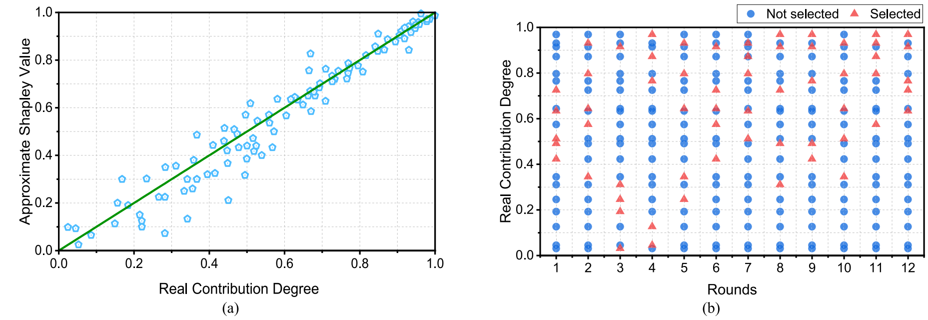


Fig. 12. Worker selection and identification analysis.

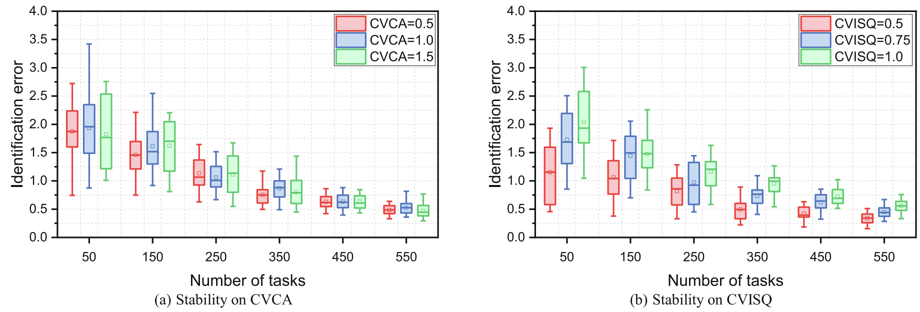


Fig. 13. Stability analysis.

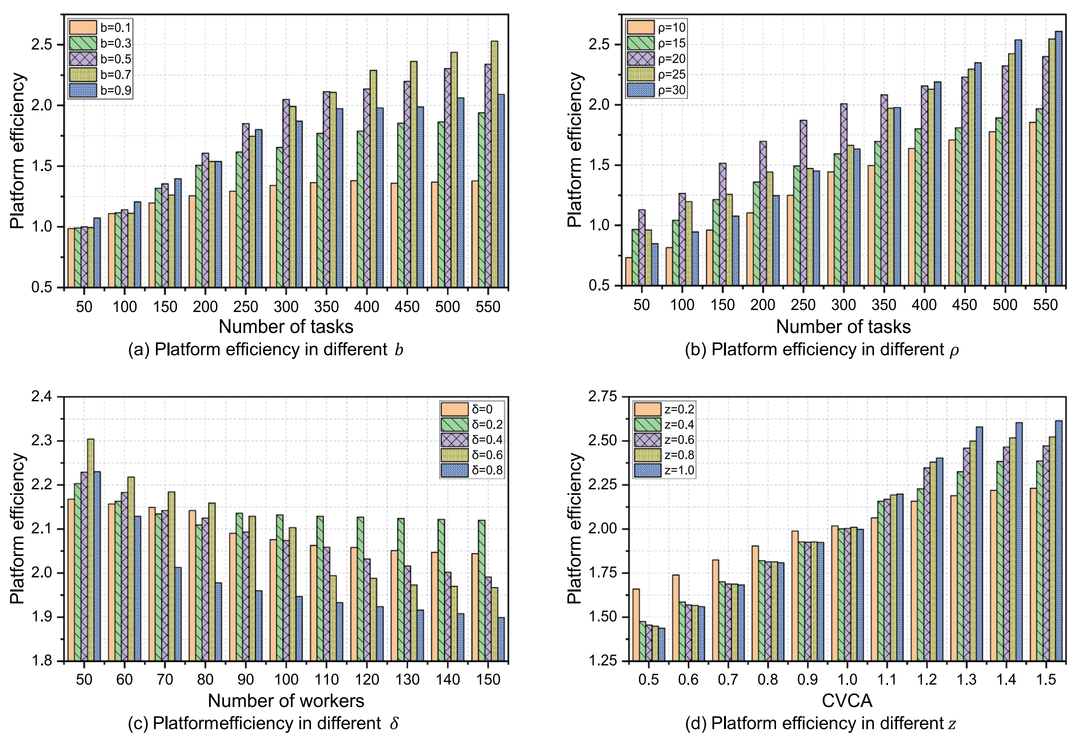


Fig. 14. Hyperparameters analysis.

depicted in Fig. 14(a). In scenarios with fewer tasks, different b in the IICA-DWS algorithm showcased relatively similar platform efficiency due to the smaller γ. As the number of tasks ranged between 200 and 350, $b = 0.5$ demonstrated superior performance. It showed an approximate 15.6 % improvement over other b on average. However, with the number of tasks within the [400, 550] range, $b = 0.7$ exhibited notably better results. It showed an approximate 17.1 % improvement over other b on average. The optimal b for achieving the highest platform efficiency varied across different numbers of tasks. According to Eqs. (13–14), b controls the update rate of ASVs and weighted sensing qualities: lower b speeds up their updates, while higher value slows them down. A smaller b might make the platform overly emphasize current sensing results but neglect workers' historical performance. Thus, it results in inaccurate worker evaluation. Conversely, a larger b might cause the platform to overlook shifts in workers' abilities promptly. As the regulatory factor increases, ASV plays a more significant role in the platform's evaluation strategy. Selecting an appropriate b is important as it not only facilitates the reasonable update of workers' ASVs and weighted sensing qualities but also ensures that the platform does not overly emphasize historical or current sensing results.

The initialization speed of regulatory factor: In the dynamic evaluation system of the IICA-DWS algorithm, the regulatory factor γ is of great importance. According to Eq. (17), when there is low confidence in ASVs, adjusting the hyperparameter ρ can effectively alter the initialization speed of γ, thereby impacting the evaluation strategy of the IICA-DWS algorithm. We conducted experiments within the range of [10, 30] for ρ, increasing it by 5 in each trial. The results are illustrated in Fig. 14(b). When the number of tasks was between 50 and 350, ρ=20 demonstrated notably better platform efficiency. It showed about a 27.4 % improvement over other ρ on average. However, when the number of tasks exceeded 450, ρ=30 exhibited a more pronounced improvement in platform efficiency. On average, it showed approximately an 18.8 % improvement over other ρ. This is because a smaller ρ leads the platform to lean toward assessing workers' ASVs without sufficient confidence at the lower number of tasks, thereby compromising the accuracy of worker evaluations. Conversely, a higher ρ may result in excessive exploration of ASVs. With more tasks completed, a larger ρ assists the platform in gaining a more comprehensive knowledge of workers' abilities and allows it to make more accurate decisions.

The exploration for recruiting workers: According to Eqs. (22–23), the IICA-DWS algorithm selects workers based on the UCB indexes. The hyperparameter δ controls the additive factor. It grants workers with less task participation more opportunities for engagement in tasks and aids the platform in identifying outstanding workers. As shown in Fig. 14(c), we conducted experiments within the range of [0, 0.8] for δ and incremented it by 0.2 in each trial. $\delta = 0$ means the platform does not offer any additional opportunities to workers with less task participation. Thus, it makes the system more susceptible to randomness. A higher δ indicates the platform offers these workers more chances for task participation. However, the experimental results reveal that the impact of δ on platform efficiency is not monotonic. For instance, with a smaller team of 50 workers, the platform had more opportunities to explore workers additionally with the same number of tasks. Consequently, $\delta = 0.6$ exhibited the highest platform efficiency. It showed an approximate improvement compared to $\delta = 0$ by 6.6 %. Yet, with a larger team of 150 workers, the platform invested more resources in the initial exploration of workers. In our simulations, a smaller $\delta = 0.2$ enabled the platform to allocate more resources to the exploitation phase and resulted in higher platform efficiency. Specifically, $\delta = 0.2$ yielded an approximately 11.6 % improvement compared to $\delta = 0.8$ in this case.

The upper limit of the regulatory factor: In MCS systems with various types of tasks, the level of collaboration among workers may differ significantly. Therefore, in cases where collaboration is weaker, relying excessively on ASV for worker evaluation might lead the platform to be more vulnerable to random factors. And it might decrease accuracy in worker evaluation. Thus, we defined the hyperparameter z in Eq. (16) to control the upper limit of the proportion of the regulatory factor in evaluating workers' comprehensive abilities. We conducted experiments within the range of [0.2, 1.0] for z and increased it by 0.2 in each trial. The strength of collaboration among workers within the team was represented by CVCA, as depicted in Fig. 14(d). From the experimental results, it is evident that with lower CVCAs, a smaller z reduced the proportion of ASVs in evaluating workers' comprehensive abilities, contributing to increased platform efficiency. Conversely, when CVCA is higher, signifying significant collaboration within the team, a larger z helped the platform fully leverage this collaborative relationship, thereby enhancing platform efficiency. Specifically, when the CVCA ranged between 0.5 and 0.9, $z = 0.2$ achieved the highest platform efficiency, averaging an improvement of 7.8 % over other z. Meanwhile, when the CVCA ranged between 1.3 and 1.5, $z = 1.0$ yielded the best platform efficiency, showing an average improvement of 8.7 % over other z.

The analysis of these hyperparameters in the IICA-DWS algorithm reflects their diverse functions. Thus, the platform can adjust these parameters based on specific circumstances. Moreover, the platform can dynamically update certain hyperparameters to pursue higher platform efficiency.

5.4. Engineering applications

In fields such as climate, environment, and transportation, as shown in Fig. 15, the platform needs to recruit workers to complete various tasks. With the development of information technology, it is possible to use crowds' mobile devices to collect a significant amount of spatiotemporal data, which makes MCS a feasible data-collecting method. However, most MCS solutions do not consider the collaborative relationships among workers. Thus, we proposed the IICA-DWS algorithm to assess the workers' individual and collaborative abilities and build teams with high efficiency. Consequently, the platform can acquire high-quality data at low cost.

In terms of the climate, we can use the IICA-DWS algorithm to recruit workers to measure the composition of the atmosphere efficiently and get a wide range of spatiotemporal data. Thus, we are able to evaluate and predict the climate of a certain area. Furthermore, we can not only make the weather forecast more accurate but also predict the disasters caused by the climate to reduce the effect of severe weather.

In terms of the traffic, we can use MCS to collect data on the traffic flow in different spatiotemporal distributions and use IICA-DWS algorithm to improve the sensing efficiency. Thus, an effective traffic flow monitoring mechanism can be established. This supports road navigation, travel guides, parking selection, and so on.

In terms of the environment, the IICA-DWS algorithm can assist us in recruiting workers to detect the spatiotemporal distribution of information such as soil composition and water quality. Thus, it enables us to establish a powerful environment monitoring system, enhance source management, and achieve feasible solutions to deal with pollution.

In addition, the IICA-DWS algorithm has many potential practical applications. In addition to MCS, it may also play a role in problems such as Internet of Vehicles and Federated Learning, making the process of information collection and processing more efficient.

Fig. 15. Applications of IICA-DWS.

6. Conclusion

In various fields, it is necessary to collect distributed data, and MCS systems play a crucial role. In this work, we consider the collaborative relationships among workers and have designed an effective algorithm for dynamic worker evaluation and team formation named IICA-DWS.

The IICA-DWS algorithm provides a dynamic worker evaluation approach called IICAE, which computes the comprehensive abilities of workers within a team. Specifically, the IICA-DWS algorithm offers an approximate method for calculating workers' contributions. Simultaneously, the IICAE approach can adjust the platform's evaluation of workers in various situations to approximate optimal results. Moreover, the IICA-DWS algorithm forms teams by considering workers' comprehensive sensing abilities and social situations. Thus, we can improve the collaboration within teams and achieve higher platform efficiency. Its team formation process adopts an MAB-based model and balances the platform's exploration and exploitation of workers. It also ensures the effectiveness of worker evaluations, platform rationality, and fairness.

Finally, extensive simulation experiments conducted on real datasets demonstrate that our algorithm can select more outstanding workers and significantly improve platform efficiency and data quality compared to other algorithms. Additionally, when collaboration within teams is more apparent, the IICA-DWS algorithm exhibits considerable improvements. The IICA-DWS algorithm also provides theoretical guarantees on feasibility and can be easily popularized.

CRediT authorship contribution statement

Yaohui Han: Methodology, Software, Visualization, Writing – original draft. **Mingyang Zhao:** Software, Visualization. **Nuanqiao Shan:** Writing – review & editing, Investigation. **Anfeng Liu:** Conceptualization, Supervision. **Tian Wang:** Validation. **Houbing Song:** Writing – review & editing. **Shaobo Zhang:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (62072475).

References

Abououf, M., Mizouni, R., Singh, S., Otrok, H., & Ouali, A. (2019). Multi-worker multi-task selection framework in mobile crowd sourcing. *Journal of Network and Computer Applications, 130*, 52–62.

Al-qaness, M. A., Dahou, A., Abd Elaziz, M., & Helmi, A. M. (2022). Multi-ResAtt: Multilevel residual network with attention for human activity recognition using wearable sensors. *IEEE Transactions on Industrial Informatics, 19*(1), 144–152.

Azzam, R., Mizouni, R., Otrok, H., Singh, S., & Ouali, A. (2018). A stability-based group recruitment system for continuous mobile crowd sensing. *Computer Communications, 119*, 1–14.

Azzam, R. H. (2016). *A Group-based Recruitment System for Mobile Crowd Sensing.*

Bai, J., Gui, J., Huang, G., Zhang, S., & Liu, A. (2023). UAV-supported intelligent truth discovery to achieve low-cost communications in mobile crowd sensing. *Digital Communications and Networks*.

Cai, K., Liu, X., Chen, Y. Z. J., & Lui, J. C. (2022). Learning with guarantee via constrained multi-armed bandit: Theory and network applications. *IEEE Transactions on Mobile Computing.*

Cai, Z., Duan, Z., & Li, W. (2020). Exploiting multi-dimensional task diversity in distributed auctions for mobile crowdsensing. *IEEE Transactions on Mobile Computing, 20*(8), 2576–2591.

Campana, M. G., & Delmastro, F. (2022). On-device modeling of user's social context and familiar places from smartphone-embedded sensor data. *Journal of Network and Computer Applications, 205*, Article 103438.

Chalkiadakis, G., Elkind, E., & Wooldridge, M. (2022). *Computational aspects of cooperative game theory*. Springer Nature.

Chen, H., Lundberg, S. M., & Lee, S. I. (2022). Explaining a series of models by propagating Shapley values. *Nature Communications, 13*(1), 4512.

Cheng, Y., Li, X., Li, Z., Jiang, S., Li, Y., Jia, J., & Jiang, X. (2014, November). AirCloud: A cloud-based air-quality monitoring system for everyone. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems* (pp. 251-265).

Estrada, R., Mizouni, R., Otrok, H., Ouali, A., & Bentahar, J. (2017). A crowd-sensing framework for allocation of time-constrained and location-based tasks. *IEEE Transactions on Services Computing, 13*(5), 769–785.

Fathian, M., Saei-Shahi, M., & Makui, A. (2017). A new optimization model for reliable team formation problem considering experts' collaboration network. *IEEE Transactions on Engineering Management, 64*(4), 586–593.

Fu, X., Liu, A., Xiong, N. N., Wang, T., & Zhang, S. (2024). ATWR-SMR: An area-constrained truthful worker recruitment based sensing map recovery scheme for sparse MCS in extreme-environment internet-of-things. *IEEE Internet of Things Journal, 11*(3), 3711–3724.

Gad-Elrab, A., & Noaman, A. Y. (2022). Clustering Ant Colony-Based Edge-Server Location Strategy in Mobile Crowdsensing. *Applied Computational Intelligence and Soft Computing, 2022.*

Gao, G., Huang, H., Xiao, M., Wu, J., Sun, Y. E., & Du, Y. (2021). Budgeted unknown worker recruitment for heterogeneous crowdsensing using cmab. *IEEE Transactions on Mobile Computing, 21*(11), 3895–3911.

"Geograph: Geographically representative photographs of the whole great britain and ireland", Mar. 2020, [online] Available: http://www.geograph.org.uk/.

Jiang, J., Di, K., An, B., Jiang, Y., Bu, Z., & Cao, J. (2022). Batch crowdsourcing for complex tasks based on distributed team formation in E-markets. *IEEE Transactions on Parallel and Distributed Systems, 33*(12), 3600–3615.

Karaliopoulos, M., & Bakali, E. (2020). Optimizing mobile crowdsensing platforms for boundedly rational users. *IEEE Transactions on Mobile Computing, 21*(4), 1305–1318.

Karaliopoulos, M., Telelis, O., & Koutsopoulos, I. (2015). User recruitment for mobile crowdsensing over opportunistic networks. In *2015 IEEE Conference on Computer Communications (INFOCOM)* (pp. 2254–2262). IEEE.

Liao, Z., Xu, X., Fan, X., Zhang, Y., & Yu, S. (2021). GRBMC: An effective crowdsourcing recommendation for workers groups. *Expert Systems with Applications, 179*, Article 115039.

Liu, Q., Luo, T., Tang, R., & Bressan, S. (2015). An efficient and truthful pricing mechanism for team formation in crowdsourcing markets. In *2015 IEEE International Conference on Communications (ICC)* (pp. 567–572). IEEE.

Liu, R., Xie, M., Liu, A., & Song, H. (2024). Joint Optimization Risk Factor and Energy Consumption in IoT Networks With TinyML-Enabled Internet of UAVs. *IEEE Internet of Things Journal.* https://doi.org/10.1109/JIOT.2023.3348837

Liu, Y., Liu, A., Wang, T., Liu, X., & Xiong, N. N. (2019). An intelligent incentive mechanism for coverage of data collection in cognitive Internet of Things. *Future Generation Computer Systems, 100*, 701–714.

Lu, J., Qu, Z., Liu, A., Zhang, S., & Xiong, N. N. (2024). MLM-WR: A Swarm Intelligence-based Cloud-Edge-Terminal Collaboration Data Collection Scheme in The Era of AIoT. *IEEE Internet of Things Journal, 11*(1), 243–255.

Lykourentzou, I., Wang, S., Kraut, R. E., & Dow, S. P. (2016, May). Team dating: A self-organized team formation strategy for collaborative crowdsourcing. In *Proceedings of the 2016 CHI conference extended abstracts on human factors in computing systems* (pp. 1243-1249).

Mohan, P., Padmanabhan, V. N., & Ramjee, R. (2008, November). Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *Proceedings of the 6th ACM conference on Embedded network sensor systems* (pp. 323-336).

Nguyen, T. N., & Zeadally, S. (2021). Mobile crowd-sensing applications: Data redundancies, challenges, and solutions. *ACM Transactions on Internet Technology (TOIT), 22*(2), 1–15.

Niwattanakul, S., Singthongchai, J., Naenudorn, E., & Wanapu, S. (2013, March). Using of Jaccard coefficient for keywords similarity. In *Proceedings of the international multiconference of engineers and computer scientists* (Vol. 1, No. 6, pp. 380-384).

Ouyang, Y., Huang, G., Zeng, Z., Wang, T., & Liu, A. (2023). TSPR: A Trusted Service Prerequest System for Efficient Data Processing in C-ITS. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 53*(12), 7364–7377.

Pan, Z., Yu, H., Miao, C., & Leung, C. (2016, March). Efficient collaborative crowdsourcing. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 30, No. 1).

Peng, Y., Liu, W., Liu, A., Wang, T., Song, H., & Zhang, S. (2024). Truth based three-tier Combinatorial Multi-Armed Bandit ecosystems for mobile crowdsensing. *Expert Systems with Applications, 236*, Article 121119.

Ren, Y., Liu, Y., Zhang, N., Liu, A., Xiong, N. N., & Cai, Z. (2018). Minimum-cost mobile crowdsourcing with QoS guarantee using matrix completion technique. *Pervasive and Mobile Computing, 49*, 23–44.

Sensorly. (2021, Jun.) *Sensorly*. [Online]. Available: http://www.sensorly.com.

Shapley, L. S. (1953). *A value for n-person games.*

Tang, J., Fan, K., Yin, P., Qu, Z., Liu, A., Xiong, N. N., & Zhang, S. (2023). DLFTI: A Deep Learning based Fast Truth Inference Mechanism for Distributed Spatiotemporal Data in Mobile Crowd Sensing. *Information Sciences, 119245*.

Tang, J., Han, F., Fan, K., Xie, W., Yin, P., Qu, Z., & Wang, T. (2023). Credit and quality intelligent learning based multi-armed bandit scheme for unknown worker selection in multimedia MCS. *Information Sciences, 647*, Article 119444.

Thiagarajan, A., Ravindranath, L., LaCurts, K., Madden, S., Balakrishnan, H., Toledo, S., & Eriksson, J. (2009). November). Vtrack: Accurate, energy-aware road traffic delay

estimation using mobile phones. In *Proceedings of the 7th ACM conference on embedded networked sensor systems* (pp. 85–98).
Tian, H., Sheng, W., Shen, H., & Wang, C. (2020). Truth finding by reliability estimation on inconsistent entities for heterogeneous data sets. *Knowledge-Based Systems, 187*, Article 104828.
Wang, E., Yang, Y., Wu, J., Liu, W., & Wang, X. (2017). An efficient prediction-based user recruitment for mobile crowdsensing. *IEEE Transactions on Mobile Computing, 17*(1), 16–28.
Wang, H., Yang, Y., Wang, E., Liu, W., Xu, Y., & Wu, J. (2022). Truthful user recruitment for cooperative crowdsensing task: A combinatorial multi-armed bandit approach. *IEEE Transactions on Mobile Computing*.
Wang, L., Liu, A., Xiong, N. N., Zhang, S., Wang, T., & Dong, M. (2024). SD-SRF: An Intelligent Service Deployment Scheme for Serverless-operated Cloud-Edge Computing in 6G Networks. *Future Generation Computer Systems, 151*, 242–259.
Wang, L., Yu, Z., Han, Q., Yang, D., Pan, S., Yao, Y., & Zhang, D. (2020). Compact scheduling for task graph oriented mobile crowdsourcing. *IEEE Transactions on Mobile Computing, 21*(7), 2358–2371.
Wang, W., Jiang, J., An, B., Jiang, Y., & Chen, B. (2016). Toward efficient team formation for crowdsourcing in noncooperative social networks. *IEEE transactions on cybernetics, 47*(12), 4208–4222.
Wu, D., Guan, P., Zhang, P., Yang, Z., & Wang, R. (2023). Task Allocation Method of Mobile Crowdsensing Based on Group Collaboration. *Journal of Electronics & Information Technology, 45*(7), 1–9.
Xie, H., & Lui, J. C. (2022). *Cooperation Preference Aware Shapley Value: Modeling, Algorithms and Applications.* IEEE/ACM Transactions on Networking.
Xu, J., Luo, Z., Guan, C., Yang, D., Liu, L., & Zhang, Y. (2022). Hiring a team from social network: Incentive mechanism design for two-tiered social mobile crowdsourcing. *IEEE Transactions on Mobile Computing*.
Yang, Q., & Wang, H. (2015). Toward trustworthy vehicular social networks. *IEEE Communications Magazine, 53*(8), 42–47.
Yang, X., Zeng, Z., Liu, A., Xiong, N. N., Wang, T., & Zhang, S. (2023). A Decentralized Trust Inference Approach with Intelligence to Improve Data Collection Quality for Mobile Crowd Sensing. *Information Sciences, 119286*.
Yin, B., Liu, Y., Xu, B., Chen, H., & Tang, S. (2023). TTAF: A two-tier task assignment framework for cooperative unit-based crowdsourcing systems. *Journal of Network and Computer Applications, 218*, Article 103719.
Yin, B., Lu, J., & Wei, X. (2021). Correlation-based task processing plans in crowdsensing platforms. *IEEE Transactions on Network Science and Engineering, 8*(2), 1542–1556.
Yu, J., Liu, A., Xiong, N. N., Zhang, S., Wang, T., & Dong, M. (2023). Employing Social Participants for Timely Data Collection Using Pub/sub Solutions in Dynamic IoT Systems. *Computer Networks, 220*, Article 109501.
Yucel, F., & Bulut, E. (2020). User satisfaction aware maximum utility task assignment in mobile crowdsensing. *Computer Networks, 172*, Article 107156.
Yucel, F., Yuksel, M., & Bulut, E. (2020). QoS-based budget constrained stable task assignment in mobile crowdsensing. *IEEE Transactions on Mobile Computing, 20*(11), 3194–3210.
Zhang, R., Li, Z., Xiong, N. N., Zhang, S., & Liu, A. (2022). TDTA: A truth detection based task assignment scheme for mobile crowdsourced Industrial Internet of Things. *Information Sciences, 610*, 246–265.
Zhao, P., Li, X., Gao, S., & Wei, X. (2022). Cooperative task assignment in spatial crowdsourcing via multi-agent deep reinforcement learning. *Journal of Systems Architecture, 128*, Article 102551.