Online Worker Scheduling for Maximizing Long-Term Utility in Crowdsourcing with Unknown Quality

JIAJUN WANG, College of Computer Science, Beijing University of Technology, Beijing, China
PENGFEI LIN, College of Computer Science, Beijing University of Technology, Beijing, China
XINGJIAN DING, College of Computer Science, Beijing University of Technology, Beijing, China
JIANXIONG GUO, Advanced Institute of Natural Sciences, Beijing Normal University, Zhuhai, China and Guangdong Key Lab of AI and Multi-Modal Data Processing, BNU-HKBU United International College, Zhuhai, China
ZHIQING TANG, Advanced Institute of Natural Sciences, Beijing Normal University, Zhuhai, China
DEYING LI, School of Information, Renmin University of China, Beijing, China
WEILI WU, Department of Computer Science, The University of Texas at Dallas, Richardson, United States

Spatiotemporal Mobile CrowdSourcing (MCS) is a new intelligent sensing paradigm for large-scale data acquisition where requesters can recruit a crowd of workers to perform data collection tasks. How to recruit suitable workers in a dynamic environment to maximize platform utility is a key issue and has become a research hotspot. Many past studies have made great efforts in this regard, but most of them either assume that the worker quality is known in advance or ignore the limitations of workers' short-term ability to provide resources. In this article, we consider a platform-centered online spatiotemporal MCS system where mobile workers have both long-term and short-term constraints for providing resources, and their quality is unknown to the platform, while the platform has a long-term budget constraint for recruiting workers. We aim to find an online worker scheduling scheme to maximize the platform's long-term utility without violating the constraints of both workers and the platform. To address this problem, we first transform the long-term utility maximization problem into a real-time utility maximization problem by leveraging the

This work is an extended version of the paper in Reference [34], which has been published at the 13th IEEE/CIC International Conference on Communications in China (ICCC'24), Hangzhou, China.
This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants No. 62202055 and No. 62202016, the Start-up Fund from Beijing Normal University under Grant No. 312200502510, the Internal Fund from BNU-HKBU United International College under Grant No. UICR0400003-24, the Project of Young Innovative Talents of Guangdong Education Department under Grant No. 2022KQNCX102, the National Science Foundation (NSF) under Grants No. 1907472 and No. 1822985, and the Interdisciplinary Intelligence SuperComputer Center, Beijing Normal University (Zhuhai).
Authors' Contact Information: Jiajun Wang, College of Computer Science, Beijing University of Technology, Beijing, Beijing, China; email: S202375129@emails.bjut.edu.cn; Pengfei Lin, College of Computer Science, Beijing University of Technology, Beijing, Beijing, China; e-mail: linpengfei@bjut.edu.cn; Xingjian Ding, College of Computer Science, Beijing University of Technology, Beijing, Beijing, China; e-mail: dxj@bjut.edu.cn; Jianxiong Guo (Corresponding author), Advanced Institute of Natural Sciences, Beijing Normal University, Zhuhai, Guangdong, China and Guangdong Key Lab of AI and Multi-Modal Data Processing, BNU-HKBU United International College, Zhuhai, Guangdong, China; e-mail: jianxiongguo@bnu.edu.cn; Zhiqing Tang, Advanced Institute of Natural Sciences, Beijing Normal University, Zhuhai, Guangdong, China; e-mail: zhiqingtang@bnu.edu.cn; Deying Li, School of Information, Renmin University of China, Beijing, Beijing, China; e-mail: deyingli@ruc.edu.cn; Weili Wu, Department of Computer Science, The University of Texas at Dallas, Richardson, Texas, United States; e-mail: weiliwu@utdallas.edu.

ACM 1533-5399/2025/2-ART4
https://doi.org/10.1145/3708893

Lyapunov optimization, then design algorithms based on the Upper Confidence Bound (UCB) and Markov approximation to solve each real-time utility maximization problem with unknown worker quality. We demonstrate that our UCB-based algorithm has a sublinear regret and prove that our proposed framework has a performance guarantee for the addressed problem. Finally, we evaluate our design through numerical simulation experiments, and the results demonstrate the effectiveness of our algorithm.

CCS Concepts: • **Networks** → **Network algorithms**; **Network economics**; **Mobile networks;**

Additional Key Words and Phrases: Mobile crowdsourcing, unknown worker scheduling, sensing quality, recruitment cost minimization, and profit maximization

ACM Reference Format:
Jiajun Wang, Pengfei Lin, Xingjian Ding, Jianxiong Guo, Zhiqing Tang, Deying Li, and Weili Wu. 2025. Online Worker Scheduling for Maximizing Long-Term Utility in Crowdsourcing with Unknown Quality. *ACM Trans. Internet Technol.* 25, 1, Article 4 (February 2025), 25 pages. https://doi.org/10.1145/3708893

1 Introduction

Smart devices and sensor networks are widely deployed in modern society. These deployments span a variety of application scenarios, including smart manufacturing, smart cities, and energy efficiency optimization [10, 16, 19, 37]. Notably, these applications often generate many sensing tasks requiring real-time and efficient processing under limited resources. To meet these stringent demands, spatiotemporal **Mobile Crowdsourcing (MCS)** has emerged as a viable solution, in which a crowd of workers is scheduled to complete sensing tasks with their smart devices when they visit some pre-designated places [17, 35, 39].

In a typical spatiotemporal MCS system, the platform publishes tasks online [32], then schedules appropriate mobile workers to perform the corresponding tasks, and specifies the amount of resources the workers need to provide. The utility achieved by the platform is related to the workers' quality of task execution and the amount of resources provided by workers. However, due to the differences in sensing devices and behaviors among workers, the sensing qualities vary for workers, even when performing the same task. In addition, it is not feasible for workers to evaluate their own quality, because it is impossible to judge the evaluation criteria and honesty of the workers. Therefore, the quality of workers is often unknown to the platform, and how to efficiently schedule workers with unknown quality is crucial to improving the utility of the platform. There are lots of works focusing on the worker scheduling problem, but most of them assume that the quality of workers is known in advance [11, 18, 21], which is unrealistic in many practical scenarios [14, 31]. Although few works have also focused on the problem of worker scheduling with unknown quality [8, 15, 30, 41, 46], they only consider the case that tasks are homogeneous, which is not suitable for the general scenario.

Generally, both the worker's resources for task execution and the platform's budget to pay for the workers are limited in the long term. For instance, in smart cities, workers equipped with mobile devices are recruited to report real-time traffic conditions. However, their resources are limited in the long run, such as the limited battery power of the carried devices or the limited time that workers can perform tasks. For the smart city platform, the long-term budget for recruiting workers is also limited. In addition to long-term constraints, workers also have certain short-term constraints when performing tasks. For example, due to equipment limitations, the amount of resources workers can provide in the short term is very constrained, and they can only perform a small number of tasks. Therefore, maximizing the long-term utility of the platform under short-term and long-term constraints on resources and budgets is a pressing issue. Current research on maximizing platform utility mainly focuses on satisfying long-term constraints while ignoring workers' short-term resource provision capacity constraints [2, 6, 12, 26].

In this article, we investigate the online worker scheduling problem to maximize the platform's long-term utility for the spatiotemporal MCS system with unknown worker quality, where we consider both long-term and short-term constraints of workers' resource provision capacity, as well as the long-term constraint of the platform's budget. As the quality of workers is unknown for the platform, an intuitive approach is to let the platform schedule workers to perform some tasks in the initial rounds and evaluate their quality, which can be called "exploration." Then, the platform could schedule workers with higher qualities based on the learned quality information to achieve greater utility, which can be called "exploitation." Our objective is to design an online worker scheduling scheme to maximize the platform's long-term utility under long-term and short-term constraints of workers' resource provision capacity while ensuring that the platform's long-term budget constraint is not exceeded.

There are three main challenges in achieving this goal. First, too much exploration will waste the platform budget to a certain extent, while too little exploration will lead to an inaccurate estimation of workers' quality; therefore, determining a better trade-off between exploration and exploitation phases has a significant impact on maximizing the utility of the platform. Second, as both workers and the platform have long-term constraints, how to allocate the amount of resources and budget to each time slot is a very crucial issue as it is almost impossible for us to get complete future information about tasks. Third, as each worker has a short-term resource constraint, how to assign tasks to corresponding workers to maximize the total utility is very tough as it falls into the category of the multi-knapsack problem.

To address the challenges mentioned above, we model the worker scheduling problem with unknown quality as a **Multi-Armed Bandit (MAB)** problem to tackle the trade-off between exploration and exploitation. Each worker can be regarded as an arm of MAB, and the reward is set to the quality of the corresponding worker. The entire worker scheduling process is formulated as a combinatorial arm-pulling process. To address the arm-pulling problem, we use the pre-defined **Upper Confidence Bound (UCB)** index to greedily select the arms to be pulled, and the index is continuously updated during the worker scheduling process. For the long-term constraints of the worker's resources and the platform's budget, we apply Lyapunov optimization techniques to transform the long-term constraints into real-time constraints, converting the original problem into a queue stability control problem. On this basis, we incorporate workers' short-term endurance as a short-term constraint on worker resources into the problem. We employ the Markov approximation to solve the queue stability control problem. Based on the Markov approximation and UCB, we propose the scheduling algorithm for workers with unknown quality.

This work is a journal extension to our previous conference paper [34]. Compared to Reference [34], we have significantly revised and clarified the paper, and improved many technique details. The primary improvements can be summarized as follows. First, we further consider the situation in that worker's quality is unknown (Section 3). To address the challenges posed by the newly formulated problem, we propose a novel worker scheduling algorithm with unknown quality (i.e., Algorithm 3). Additionally, we prove the performance upper bound of the proposed algorithm and conducted regret analysis on the results of learning workers' task completion quality (Section 5). Finally, we thoroughly re-implement the simulations to verify our designs (Section 6). The contributions of our work are summarized as follows:

— We investigate the online worker scheduling problem with unknown worker quality for the spatiotemporal MCS system. Different from existing studies that only consider long-term constraints, we also consider short-term constraints of workers' resource provision capacity, which is a relatively underexplored topic.

— We transform the online worker scheduling problem into a real-time problem by leveraging the Lyapunov optimization and then solve the real-time problem by combining the Markov approximation algorithm and the UCB algorithm.
— We prove that our UCB-based algorithm has a limited regret, and our proposed framework has a performance guarantee for the addressed problem.
— We conduct extensive simulations to validate the performance of our designs, and the results show the effectiveness of our algorithms.

The remainder of this article is structured as follows: Section 2 reviews the related works of worker scheduling. Section 3 introduces the system model and problem formulation. Section 4 introduces the proposed online worker scheduling framework. Section 5 analyzes the performance of our algorithms. Section 6 presents the simulations. At last, Section 7 concludes this article.

2 Related Works

In this section, we briefly review the research efforts on worker scheduling with unknown quality and online worker scheduling with long-term constraints.

2.1 Worker Scheduling with Unknown Quality

MAB is an effective reinforcement learning model for making online decisions in unknown information environments. In recent years, the application of MAB in scheduling workers with unknown task completion quality in spatiotemporal MCS systems has been widely studied [13, 15, 27]. For instance, Li et al. [14] study the worker recruitment problem under the uncertainty of workers dynamically joining and leaving, and propose a context-aware online CMAB-based incentive mechanism. Gao et al. [8] model the worker recruitment process with unknown task completion quality as a CMAB problem, aiming to develop a recruitment strategy that maximizes the total weighted task completion quality under a limited budget, in which hey also consider the case that the worker recruitment costs are unknown. To address the problem, they propose an unknown-worker recruitment algorithm based on the modified UCB algorithm. Wang et al. [33] study the heterogeneous worker recruitment problem by considering both worker ability and subjective collaboration. They propose a CMAB-based multi-round user recruitment strategy under budget constraints and use a graph theory-based algorithm to improve the issue of requiring a fixed number of workers in each round of recruitment. Xiao et al. [41] design a joint optimization method for platform utility and task completion quality for the task allocation problem with unknown task completion quality. Zhang et al. [45] study the worker recruitment problem with unknown worker quality in the context of edge computing and spatiotemporal MCS systems, and propose a CMAB-based user recruitment algorithm for the problem. Peng et al. [25] study the task allocation problem where there are multiple service platforms and the worker quality is unknown. They propose a multi-platform collaborative task allocation mechanism based on MAB. Fu et al. [7] study the situation where both worker quality and the task publisher's payment are unknown. They propose a truthful and dual-directional CMAB scheme, in which they utilize the UCB algorithm to maximize the total profit for both the workers and the task publishers. However, most of these works assume that the tasks performed by workers are homogeneous, without considering the heterogeneity of tasks in spatiotemporal MCS systems.

2.2 Online Worker Scheduling with Long-term Constraints

In online scheduling scenarios, tasks usually arrive dynamically over time. If future task information can be obtained, then the budget or resource allocation strategy can be adjusted in time to achieve the optimal goal. However, it is basically impossible to obtain future information.

In recent years, many works have focused on the online worker scheduling problem with long-term constraints. Yang et al. [44] utilize the fuzzy time-series analysis method to predict the number of participants available for each task in a specific time and space. Then, according to the predicted results and comprehensively considering various attributes of the participants, they design an online task allocation algorithm based on the improved genetic algorithm. Peng et al. [24] study the online task allocation problem of maximizing the total profit of the mobile crowdsensing platform while meeting the time window requirements of each task. Due to the influence of the task assignment time point, there may be a reduction in task completion rates and budget utilization. To address this issue, Ding et al. [5] propose a dynamic delayed-decision task assignment method, which first considers the task assignment time point and introduces a model for assigning multiple tasks under spatiotemporal constraints. Then, they propose two mobility prediction methods to efficiently compute the probabilities of users reaching the task destinations. Additionally, some research has focused on developing task allocation strategies by establishing appropriate incentive mechanisms. Gao et al. [9] focus on UAV-assisted MCS scenarios and propose a UAV-assisted multi-task allocation method to optimize sensing coverage and data quality. The authors aim to incentivize participants to contribute sensing data from nearby points of interest within a limited budget, in which they jointly consider optimizing task assignment and trajectory scheduling. Xu et al. [42] integrate the **graph attention networks (GAT)** with **deep reinforcement learning (DRL)**, and develop a GAT-based DRL method to address the NP-hard task allocation problem. Compared to traditional heuristic methods, their approach leverages the flexibility and adaptability of DRL. In addition to the above research, some scholars have also used Lyapunov optimization technology to balance the long-term stability and platform utility of the spatiotemporal MCS system [20, 29, 36], which is similar to our work. However, they do not fully consider the short-term resource provision capacity constraints of workers and, therefore, cannot solve the online worker scheduling problem explored in this article.

2.3 Comparison Analysis

Table 1 summarizes the difference between our work and the representative work. To sum up, existing works often overlook the heterogeneity of tasks in spatiotemporal MCS systems when considering the unknown worker quality [7, 14, 15, 33, 45], and the studies on online worker scheduling problem usually only consider long-term constraints of workers or platform [20, 38, 40]. In our work, we consider the heterogeneity of tasks, that is, different tasks bring different utilities, even if the same amount of resources are invested. In addition, we also comprehensively consider the long-term constraints of workers and platforms, as well as the short-term resource provision capacity constraints of workers, which makes our work more in line with the actual situation of spatiotemporal MCS systems. This is also the fundamental difference between our work and previous studies.

3 System Model and Problem Formulation

3.1 System Model

In this section, we first introduce the spatiotemporal MCS system model with unknown worker quality, then formulate the problem to be addressed. The main symbols we use are listed in Table 2.

As shown in Figure 1, we consider a stable spatiotemporal MCS system that contains a fixed number of m mobile workers with unknown quality denoted by $\mathcal{W} = \{w_1, w_2, \ldots, w_m\}$. The MCS platform will periodically publish tasks to workers. To conveniently represent the periodicity of system release tasks, we assume the MCS system works in a slotted model, and the timeline is divided T time slots, *i.e.*, $\mathcal{T} = \{1, 2, \ldots, T\}$. At the beginning of each time slot $t \in \mathcal{T}$, there is a

Table 1. Comparison of Our Work and the Representative Related Work

Work	Unknown worker quality	Heterogeneity of tasks	Long-term constraints	Short-term resource provision capacity constraints of workers
[14, 41]	✓	×	✓	×
[15]	✓	×	×	✓
[2]	×	×	✓	×
[33, 43]	✓	×	×	✓
[25]	×	✓	×	✓
Ours	✓	✓	✓	✓

Table 2. Symbol Definitions

Symbol	Definition
$\mathcal{T}$	Time slots set
$\mathcal{W}$	Workers set
$\mathcal{A}_t$	Tasks set in time slot t
$\boldsymbol{R}_t$	Worker resource allocation matrix in time slot t
$\overline{B_i}$	Long-term time-average resource budget of worker w_i
b_i^{max}	Resource budget of worker w_i in a single time slot
τ	Unit resource price of workers
q_i	Expected task execution quality of w_i
q_i^t	Task completion quality of w_i in time slot t
$\tilde{q}_i^t$	Learned quality for w_i until time slot t
η_i^t	Uncertainty measure of w_i in time slot t
$r_{j,t}^{min}$	Minimal resource requirement of task a_j in time slot t
$\overline{C_{bgt}}$	Long-term time-average recruitment budget of the platform

set N_t of tasks published by the platform, which is denoted by $\mathcal{A}_t = \{a_1, a_2, \ldots, a_{N_t}\}$. We assume that there are more tasks than workers in each time slot, and each task is indivisible and thus can be executed by at most one worker, while one worker may perform multiple tasks within his/her capability.

For any worker $w_i \in \mathcal{W}$, there is a long-term time-average resource budget $\overline{B_i}$, which means that the average amount of resources invested by worker w_i in executing tasks in each time slot cannot exceed $\overline{B_i}$. In addition, considering the capability of each worker to perform tasks is limited, we assume the resource budget of the worker $w_i \in \mathcal{W}$ is b_i^{max} in a single time slot. For any task $a_j \in \mathcal{A}_t$, there is a minimal resource requirement $r_{j,t}^{min}$ (where $r_{j,t}^{min} \geq 1$), which means that the task can only be executed if the amount of resources allocated to it is more than $r_{j,t}^{min}$. We use a $M \times N_t$ allocation matrix $\boldsymbol{R}_t$ to represent the worker scheduling strategy in time slot t, where each element R_{ij}^t denotes the amount of resources that worker w_i allocates to task a_j, and $R_{ij}^t = 0$ indicates that the worker w_i does not perform the task a_j in time slot t.

We assume that each worker w_i has an expected task execution quality q_i, which is unknown to the platform, and the worker himself cannot accurately judge the quality of his task execution. Let $q_i^t \in [0, 1]$ represent the task completion quality of worker w_i in time slot t, which can be judged by the platform after w_i performs tasks. Note that q_i^t may vary in different time slots, but it follows an unknown distribution with the expected task execution quality q_i. The variability of q_i^t comes from many factors, for example, the willingness of workers to perform tasks in different time slots

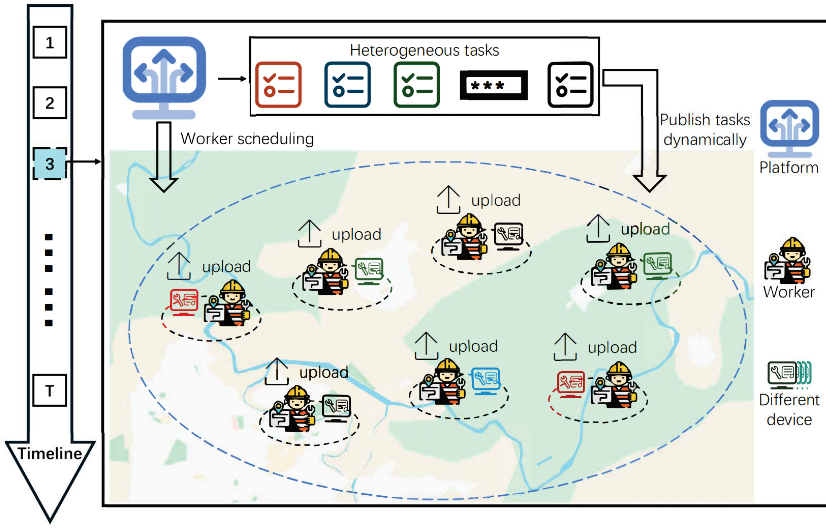


Fig. 1. The architecture of spatiotemporal MCS system.

may be different, or changes in environmental information in different time slots may also lead to changes in the quality of workers' task execution.

Let P_{ij}^t denote the profit obtained by the platform from worker w_i performing task a_j in time slot t. With the above models and assumptions, we can calculate P_{ij}^t as follows:

$$P_{ij}^t = \begin{cases} \alpha_j \log(1 + \beta_j q_i^t R_{ij}^t), & R_{ij}^t \geq r_{j,t}^{min}, \\ 0, & R_{ij}^t < r_{j,t}^{min}. \end{cases} \tag{1}$$

Assume that the unit resource price of workers is identical and is denoted as τ, then the cost of the platform for purchasing worker w_i's resources in time slot t is denoted by

$$C_i^t = \tau \cdot \sum\nolimits_{a_j \in \mathcal{A}_t} R_{ij}^t. \tag{2}$$

The long-term time-average recruitment cost of the platform is limited on the entire timeline and is represented by $\overline{C_{bgt}}$.

3.2 Constraints of Our Problem

Allocation decision constraint: As each task can be executed by at most one worker, the following constraint must be satisfied:

$$\sum\nolimits_{w_i \in \mathcal{W}} \mathbb{I}\{R_{ij}^t > 0\} \leq 1, \ \forall a_j \in \mathcal{A}_t, t \in \mathcal{T}, \tag{3}$$

where $\mathbb{I}\{R_{ij}^t > 0\}$ is the indicator function.

Worker resource consumption constraints: The Worker resource consumption must satisfy the long-term and short-term constraints:

$$\lim_{T \to \infty} \frac{1}{T} \sum\nolimits_{t=1}^{T} \sum\nolimits_{a_j \in \mathcal{A}_t} \mathbb{E}\{R_{ij}^t\} \leq \overline{B_i}, \ \forall w_i \in \mathcal{W}, \tag{4}$$

$$\sum\nolimits_{a_j \in \mathcal{A}_t} R_{ij}^t \leq b_i^{max}, \ \forall w_i \in \mathcal{W}, t \in \mathcal{T}. \tag{5}$$

Platform recruitment budget constraints: The time-average recruitment cost of the platform on the entire timeline must not exceed the budget $\overline{C_{bgt}}$:

$$\lim_{T \longrightarrow \infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{w_i \in \mathcal{W}} \mathbb{E}\{C_i^t\} \leq \overline{C_{bgt}}. \tag{6}$$

In the above constraints, the expectation function $\mathbb{E}\{\cdot\}$ is used to eliminate the influence of stochastic in the dynamic spatiotemporal MCS system.

3.3 Problem Definition

We aim to find an online worker scheduling strategy with unknown worker quality to maximize the long-term utility of the platform while satisfying the above constraints. The utility U_t of the platform in a time slot t is defined as the total profits of tasks minus the total recruitment cost on the entire timeline, that is,

$$U_t = \sum_{w_i \in \mathcal{W}} \sum_{a_j \in \mathcal{A}_t} P_{ij}^t - \sum_{w_i \in \mathcal{W}} C_i^t. \tag{7}$$

The problem is formally defined as follows.

Problem 1. **Online worker scheduling with unknown worker quality for Maximizing Platform's Long-term utility in Platform-centric spatiotemporal crowdsourcing systems (MPLP).** *Given the time slot sequence $\mathcal{T}$, the mobile worker set $\mathcal{W}$ with unknown quality q_i for each worker w_i, the task set $\mathcal{A}_t$ for each time slot $t \in \mathcal{T}$, the MPLP problem aims to find a worker scheduling strategy $\boldsymbol{R}_t$ for each time slot to maximize platform's long-term utility under constraint Equations (3)–(6), which can be written as*

$$\textbf{(}\boldsymbol{P}\textbf{1)} \max_{\boldsymbol{R}_1, \ldots, \boldsymbol{R}_T} \lim_{T \longrightarrow \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\{U_t\},$$
$$\text{s.t. Equations (3), (4), (5), (6).}$$

4 Online Worker Scheduling Framework

4.1 Overview

Our optimization problem includes challenges with long-term constraints, short-term constraints, and unknown worker quality. To solve this problem, we first transform the optimization problem under long-term constraints into a real-time queue stability control problem through the Lyapunov optimization technique (Section 4.2). Then, we propose an algorithm based on Markov approximation to solve the real-time optimization problem with short-term constraints (Section 4.3), in which we ignore the condition of unknown worker quality. To deal with the challenge of unknown worker quality, we model the worker quality prediction problem as an MAB problem and use the UCB algorithm to solve this problem (Section 4.4). Finally, we integrate the methods in Section 4.3 and Section 4.4, and design an algorithm to solve the online worker scheduling problem with unknown worker quality (Section 4.5).

4.2 Problem Transformation and Online Framework

The core challenge of the original MPLP problem $\boldsymbol{P1}$ is that the optimal worker scheduling strategy relies on future information, which is impossible to obtain due to the dynamic and stochastic property of the spatiotemporal MCS system. To address the issue, we decompose the long-term optimization object under long-term constraints into each single time slot based on Lyapunov optimization, and transform the original problem into a queue stability control problem.

For clarity, we define $E_i^R(t) = \sum_{a_j \in \mathcal{A}_t} R_{ij}^t - \overline{B_i}$ and $E^C(t) = \sum_{w_i \in \mathcal{W}} C_i^t - \overline{C_{bgt}}$. Then, we define virtual queues with initial values of 0 for each long-term constraint of Equations (4) and (6), respectively, that is,

$$Q_i^R(t+1) = \max\{Q_i^R(t) + E_i^R(t), 0\}, \forall w_i \in \mathcal{W}, \tag{8}$$

$$Q^C(t+1) = \max\{Q^C(t) + E^C(t), 0\}. \tag{9}$$

Each virtual queue represents the exceeded budget of each constraint. From Equation (8), we have

$$Q_i^R(t+1) - Q_i^R(t) \geq \sum\nolimits_{a_j \in \mathcal{A}_t} R_{ij}^t - \overline{B_i}, \forall w_i \in \mathcal{W}. \tag{10}$$

By summing the inequality on each $t \in \mathcal{T}$ and take expectation, we have

$$\frac{1}{T} \sum\nolimits_{t=1}^{T} \sum\nolimits_{a_j \in \mathcal{A}_t} \mathbb{E}\{R_{ij}^t\} - \overline{B_i} \leq \frac{\mathbb{E}\{Q_i^R(T)\}}{T}. \tag{11}$$

Therefore, to satisfy the constraint (4), we only need to satisfy the following constraint:

$$\lim_{T \longrightarrow \infty} \frac{\mathbb{E}\{Q_i^R(T)\}}{T} \leq 0, \forall w_i \in \mathcal{W}. \tag{12}$$

Similarly, constraint Equation (6) can be satisfied by meeting the following constraint:

$$\lim_{T \longrightarrow \infty} \frac{\mathbb{E}\{Q^C(T)\}}{T} \leq 0. \tag{13}$$

Constraint Equations (12) and (13) suggest that we only need to control the stability of the virtual queues to satisfy the long-term constraints. Next, we introduce the *Lyapunov function* and *one-slot conditional Lyapunov drift* [22] to stabilize the virtual queues.

Define $\Theta(t) \triangleq \{Q_1^R(t), Q_2^R(t), \ldots, Q_m^R(t), Q^C(t)\}$ as the vector of all virtual queues at time slot t, the Lyapunov function can be defined as follows:

$$L(\Theta(t)) \triangleq \frac{1}{2} \sum\nolimits_{i=1}^{m} Q_i^R(t)^2 + \frac{1}{2} Q^C(t)^2. \tag{14}$$

The one-slot conditional Lyapunov drift is defined as $\Delta(\Theta(t)) \triangleq \mathbb{E}[L(\Theta(t+1)) - L(\Theta(t))|\Theta(t)]$, which reflects the expected backlog increment of all virtual queues over one slot. Then, we leverage the Lyapunov drift-plus-penalty function to approximately solve our problem on each time slot t, and we get the following problem:

$$\textbf{(\textit{P}2)} \ \max_{\boldsymbol{R}_t} \mathbb{E}[V \cdot U_t | \Theta(t)] - \Delta(\Theta(t)), \forall t \in \mathcal{T},$$
$$\text{s.t. Equations (3), (5), (12), (13),}$$

where V is a positive weight that balances utility and virtual queue backlogs.

According to Lemma 4.6 in Reference [22], we can derive that

$$\Delta(\Theta(t)) \leq B + \sum\nolimits_{w_i \in \mathcal{W}} Q_i^R(t) E_i^R(t) + Q^C(t) E^C(t), \tag{15}$$

where B is a positive constant value for all $t \in \mathcal{T}$. Define that $\Omega(t) = \sum_{w_i \in \mathcal{W}} Q_i^R(t) E_i^R(t) + Q^C(t) E^C(t)$, then, problem ***P*2** can be approximately solved by addressing the following problem:

$$\textbf{(\textit{P}3)} \ \max_{\boldsymbol{R}_t} \mathbb{E}[V \cdot U_t - \Omega(t) | \Theta(t)], \forall t \in \mathcal{T},$$
$$\text{s.t. Equations (3), (5).}$$

The first component in the objective function of ***P*3** is about maximizing the platform's utility in each time slot, corresponding to the objective function of MPLP. The second component is about controlling the virtual queue backlogs, which reflects the exceeded budget of each time-average constraint. The positive weight V is used to adjust the trade-off between the two components.

ALGORITHM 1: Online Worker Scheduling (OWS) Algorithm

input : $\mathcal{W}$, $\mathcal{T}$, A_t for $t \in \mathcal{T}$, and control parameter V.
output: Worker scheduling strategies $\boldsymbol{R}_1^*, \ldots, \boldsymbol{R}_T^*$.

1 $Q^C(0) = 0$, $Q_i^R(0) = 0$ for each $w_i \in \mathcal{W}$;
2 **for** $t = 0$ *to* $T - 1$ **do**
3   Find the optimal solution $\boldsymbol{R}_t^*$ of $\boldsymbol{P3}$;
4   Calculate virtual queues $Q_i^R(t+1)$ and $Q^C(t+1)$ for the next time slot by Equations (8) and (9);
5 **return** $\boldsymbol{R}_1^*, \ldots, \boldsymbol{R}_T^*$;

By solving $\boldsymbol{P3}$ on each time slot t, we get a feasible solution for the original MPLP problem. The proposed online algorithm is described in Algorithm 1.

In Algorithm 1, we need to find the optimal solution of $\boldsymbol{P3}$, this requires us to learn the quality of workers beforehand. In addition, $\boldsymbol{P3}$ is an NP-hard problem due to its multi-knapsack property [3]. Therefore, in the Section 4.3, we propose a worker scheduling model with unknown quality based on the MAB model. Then, in the Section 4.4, We propose a Markov approximation method to solve $\boldsymbol{P3}$. Finally, we propose a solution algorithm to solve the problem in line 3 of Algorithm 1 in Section 4.5.

Notice that we omit constraint Equations (12) and (13) in $\boldsymbol{P3}$, the two constraints are hidden in the second component of $\boldsymbol{P3}$'s optimization objective function, and the solution obtained by Algorithm 1 can satisfy these two constraints, which will be proved in next section.

4.3 Markov Approximation Method

In this subsection, we design a Markov approximation-based algorithm to approximately solve $\boldsymbol{P3}$ for Algorithm 1, which is inspired by Reference [23]. We use $G(\boldsymbol{R}_t)$ to denote the objective function of $\boldsymbol{P3}$, then $\boldsymbol{P3}$ can be transformed into the following form:

$$
\begin{aligned}
(\boldsymbol{P4})\ \max\ & \sum\nolimits_{\boldsymbol{R}_t \in \mathcal{F}_t} p(\boldsymbol{R}_t) \cdot G(\boldsymbol{R}_t), \\
\text{s.t.}\ & \sum\nolimits_{\boldsymbol{R}_t \in \mathcal{F}_t} p(\boldsymbol{R}_t) = 1, \forall t \in \mathcal{T},
\end{aligned}
$$

where $\mathcal{F}_t$ is the collection of all feasible solutions, and $p(\boldsymbol{R}_t)$ represent the probability of the solution $\boldsymbol{R}_t$ is adopted at time slot t. Obviously, the optimal solution of $(\boldsymbol{P4})$ is to set $p(\boldsymbol{R}_t) = 1$ for $\boldsymbol{R}_t$ that maximize $G(\boldsymbol{R}_t)$.

Let $\Gamma = \frac{1}{\gamma} \sum_{\boldsymbol{R}_t \in \mathcal{F}_t} p(\boldsymbol{R}_t) \cdot \log p(\boldsymbol{R}_t)$, where γ denote a positive constant that controls the approximation ratio of the entropy term. Thus, this problem can be approximated as a log-sum-exp problem [4] as

$$
\begin{aligned}
(\boldsymbol{LSE - P4})\ \min\ & \sum\nolimits_{\boldsymbol{R}_t \in \mathcal{F}_t} p(\boldsymbol{R}_t) \cdot G(\boldsymbol{R}_t) + \Gamma, \\
\text{s.t.}\ & \sum\nolimits_{\boldsymbol{R}_t \in \mathcal{F}_t} p(\boldsymbol{R}_t) = 1, \forall t \in \mathcal{T}.
\end{aligned}
$$

The optimality gap between $\boldsymbol{LSE - P4}$ and $\boldsymbol{P4}$ is bounded by $\frac{1}{\gamma} \log |\mathcal{F}_t|$ according to Reference [4]. Actually, the problem $\boldsymbol{LSE - P4}$ converges to the problem $\boldsymbol{P4}$ when γ approaches infinity. By utilizing the Karush-Kuhn-Tucker condition [1], we can get the optimal solution of $\boldsymbol{LSE - P4}$ for any $t \in \mathcal{T}, \boldsymbol{R}_t \in \mathcal{F}_t$:

$$
p(\boldsymbol{R}_t) = \frac{\exp\left(\gamma \cdot G(\boldsymbol{R}_t)\right)}{\sum_{\widetilde{\boldsymbol{R}}_t \in \mathcal{F}_t} \exp\left(\gamma \cdot G\left(\widetilde{\boldsymbol{R}}_t\right)\right)}. \tag{16}
$$

ALGORITHM 2: Markov Approximation-based Algorithm

input : $\mathcal{W}$, $\mathcal{T}$, A_t for time slot t, $\mathcal{F}_t$, iteration number I_c.
output: The optimal strategy $\boldsymbol{R}_t^*$ in time slot t.

$\boldsymbol{R}_t^* = \emptyset$ and $G\left(\boldsymbol{R}_t^*\right) = 0$;
Randomly select $\boldsymbol{R}_t$ form $\mathcal{F}_t$;
while $I_c > 0$ **do**
  Calculate $G\left(\boldsymbol{R}_t\right)$;
  if $G\left(\boldsymbol{R}_t\right) > G\left(\boldsymbol{R}_t^*\right)$ **then**
    $\boldsymbol{R}_t^* = \boldsymbol{R}_t$;
  Select a new strategy $\boldsymbol{R}_t'$ based on the transition probability (17);
  Update $\boldsymbol{R}_t$ by $\boldsymbol{R}_t'$;
  $I_c = I_c - 1$;
return $\boldsymbol{R}_t^*$;

Then, we can find the solution for $\boldsymbol{P4}$ by choosing $\boldsymbol{R}_t$ with the maximum probability $p(\boldsymbol{R}_t)$ got from Equation (16). Next, we design a Markov chain-based algorithm to solve the problem $\boldsymbol{LSE{-}P4}$, which also returns a feasible solution for $\boldsymbol{P4}$.

The key idea of the Markov chain-based algorithm is to create a time-reversible ergodic Markov chain [4] that achieves the stationary distribution as shown in Equation (16). The constructed Markov chain should be irreducible, that is, any state is reachable from any other state. Also, the following balance equation should be satisfied: $p(\boldsymbol{R}_t) \cdot p(\boldsymbol{R}_t, \boldsymbol{R}_t') = p(\boldsymbol{R}_t')) \cdot p(\boldsymbol{R}_t'), \boldsymbol{R}_t), \forall \boldsymbol{R}_t, \boldsymbol{R}_t' \in \mathcal{F}_t$, and $\boldsymbol{R}_t \neq \boldsymbol{R}_t'$, where $\mathcal{F}_t$ is the state space, and $p(\boldsymbol{R}_t, \boldsymbol{R}_t')$ is the transition probability from state $\boldsymbol{R}_t$ to $\boldsymbol{R}_t'$. Based on Lemma 1 of Reference [4], we could construct such a Markov chain as follows. First, we treat the solution space $\mathcal{F}_t$ of $\boldsymbol{LSE - P4}$ as the state space, and the transition probability $p(\boldsymbol{R}_t, \boldsymbol{R}_t')$ for any two states $\boldsymbol{R}_t, \boldsymbol{R}_t' \in \mathcal{F}_t$ and $\boldsymbol{R}_t \neq \boldsymbol{R}_t'$ is set as follows:

$$p(\boldsymbol{R}_t, \boldsymbol{R}_t') = \rho \cdot \exp\left(\frac{\gamma}{2}\left(G\left(\boldsymbol{R}_t'\right) - G\Big(\boldsymbol{R}_t\Big)\right)\right), \tag{17}$$

where ρ is a positive constant.

The designed Markov chain-based algorithm is described in Algorithm 2. In the algorithm, we randomly choose a state, i.e., a worker scheduling strategy $\boldsymbol{R}_t$ from the solution space. Then, we constantly update the state according to the transition probabilities, thus forming a Markov chain, and iterate this process until the Markov chain converges. Note that during the iteration, the best strategy has been recorded.

When the Markov chain reaches the stationary distribution, or equivalently, satisfies the balance equation, we can get the optimal strategy. Recall that the optimality gap between $\boldsymbol{LSE - P4}$ and $\boldsymbol{P4}$ is bounded by $\frac{1}{\gamma}\log|\mathcal{F}_t|$, we can set γ as large as possible to get a better solution. Assume that the algorithm achieves convergence within I_c iterations, we need to calculate $|\mathcal{F}_t|$ transition probabilities in each iteration, then, the time complexity of Algorithm 2 is $O(I_c|\mathcal{F}_t|)$.

In this subsection, we explain how to solve the queue stability control problem using Markov approximation. However, the worker quality is unknown for the platform, which poses challenges to maximizing the platform's long-term utility. Therefore, in the next subsection, we will explore how to learn workers' quality with the UCB algorithm.

4.4 Worker Quality Learning

In the process of scheduling unknown quality workers, we need to continuously learn about the workers' quality and choose workers with as high quality as possible to maximize the long-term utility of the platform within a limited budget. This is actually an online learning and sequential decision-making problem, which is very similar to the MAB problem. The classic MAB problem is a typical exploration-exploitation problem [28]. It assumes that a gambler operates a multi-armed slot machine, and pulling the arm results in a reward that follows an unknown distribution. The gambler pulls one arm periodically based on bandit policy, and the goal is to maximize the cumulative reward by constantly experimenting and selecting arms. Therefore, we model the worker quality learning problem as an MAB problem. Specifically, we consider each worker as an arm, and its quality as the corresponding reward.

For the MAB problem, the UCB algorithm is an effective approach to balance exploration and exploitation. It balances the arm with a higher uncertain reward and the arm with a higher known reward by calculating a confidence interval for each arm and selecting the arm with the highest upper confidence bound. The upper confidence bound consists of two parts:

(1) Estimated average reward: calculated from the historical data of the currently selected arm.
(2) Uncertainty measurement: The fewer times the arm is selected, the higher the uncertainty.

By combining these two parts, the UCB algorithm can take into account arms with higher current rewards without ignoring those with fewer choices but higher potential rewards.

For clarity, we let CB_i^t denote the total resource consumed by worker w_i until time slot t, which can be expressed as

$$CB_i^t = CB_i^{t-1} + \sum\nolimits_{a_j \in \mathcal{A}_t} R_{ij}^t, \forall t \in \mathcal{T}, \tag{18}$$

where $CB_i^0 = 0$.

In our worker quality learning problem, we use $\tilde{q}_i^t$ to denote the learned quality for worker w_i until time slot t, i.e., the estimated average reward of the ith arm until time slot t, which can be calculated as

$$\tilde{q}_i^t = \frac{\tilde{q}_i^{t-1} CB_i^{t-1} + q_i^t \sum_{a_j \in \mathcal{A}_t} R_{ij}^t}{CB_i^t}, \forall t \in \mathcal{T} \backslash \{t=1\}, \tag{19}$$

where for each worker w_i, we set $\tilde{q}_i^1 = 1$.

We use η_i^t to denote the uncertainty measure. To balance exploration and exploitation more effectively, we assume the best will happen when facing uncertainty [43] and calculate η_i^t in the following way:

$$\eta_i^t = \sqrt{\frac{2 \ln \left(\sum_{w_i \in \mathcal{W}} CB_i^{t-1}\right)}{CB_i^{t-1}}}, \forall w_i \in \mathcal{W}, t \in \mathcal{T} \backslash \{t=1\}, \tag{20}$$

where we let $\eta_i^1 = 0$.

Let $\hat{q}_i^t$ denote the upper confidence bound, which is also called the UCB-based quality, and it can be calculated as follows:

$$\hat{q}_i^t = \tilde{q}_i^t + \eta_i^t, \forall w_i \in \mathcal{W}, t \in \mathcal{T}. \tag{21}$$

From Equations (20) and (21), we can observe that a decrease in CB_i^t leads to a corresponding increase in η_i^t, which means the workers who previously consumed fewer resources (i.e., been scheduled rarely) will have higher UCB-based quality, and thus are more likely to be scheduled in the following time slots. In the next subsection, we combine the UCB algorithm with the Markov approximation and design an algorithm to solve the online worker scheduling problem with unknown worker quality.

4.5 Algorithm Design

The goal of our problem is to maximize the platform's utility within a limited recruitment budget while ensuring the long-term and short-term constraints of workers, in other words, we aim to

ALGORITHM 3: UCB-based Algorithm

input : $t, \mathcal{W}, \mathcal{T}, \mathcal{A}_t$ for time slot t.
output: The optimal strategy $\boldsymbol{R}_t^*$ in time slot t.
1 $\boldsymbol{R}_t^* = \emptyset, \boldsymbol{R}_t^{s,*} = \emptyset, G\left(\boldsymbol{R}_t^*\right) = 0, \mathcal{W}_t^s = \emptyset$;
2 **if** $t == 1$ **then**
  `// Initialization exploration phase:`
3   Recruit all unknown workers, i.e., $\mathcal{W}_t^s \leftarrow W$;
4   **for** *each worker* $w_i \in \mathcal{W}$ **do**
5     $\tilde{q}_i^1 \leftarrow 1$;
6   Randomly assign a task $a_j \in \mathcal{A}_t$ and set $R_{ij}^t = r_{j,t}^{min}, R_{ij}^t \in \boldsymbol{R}_t^*$;
7   Calculate $R_{ij}^t \in \boldsymbol{R}_t^*$ for the unassigned task by Algorithm 2;
8 **else**
  `// Exploitation phase:`
9   $\mathcal{W}_t^{s'} = \emptyset, G(\boldsymbol{R}_t^{s',*}) = \emptyset$;
10   **for** *each worker* $w_i \in \mathcal{W}$ **do**
11   Calculate CB_i^t based on Equation (18);
12   Calculate $\tilde{q}_i^t$ based on Equation (19);
13   Calculate $\hat{q}_i^t$ based on Equations (20) and (21);
14   Sort the workers according to the UCB-based quality: $\hat{q}_1^t \geq \hat{q}_2^t \geq \cdots \geq \hat{q}_m^t$;
15   Select the worker with the highest $\hat{q}_i^t$ into $\mathcal{W}_t^s$; Remove the worker from $\mathcal{W}$;
16   calculate $\boldsymbol{R}_t^{s,*}$ by Algorithm 2 based on $\mathcal{W}_t^s$, and calculate $G(\boldsymbol{R}_t^{s,*})$;
17   **while** $\left|G(\boldsymbol{R}_t^{s,*}) - G(\boldsymbol{R}_t^{s',*})\right| > \epsilon$ **do**
18     $\mathcal{W}_t^{s'} \leftarrow \mathcal{W}_t^s, G(\boldsymbol{R}_t^{s',*}) \leftarrow G(\boldsymbol{R}_t^{s,*})$;
19   Select the worker with the highest $\hat{q}_i^t$ into $\mathcal{W}_t^s$; Remove the worker from $\mathcal{W}$;
20   Calculate $\boldsymbol{R}_t^{s,*}$ by Algorithm 2 based on $\mathcal{W}_t^s$, and calculate $G(\boldsymbol{R}_t^{s,*})$;
21   $\boldsymbol{R}_t^* \leftarrow \boldsymbol{R}_t^{s,*}$;
22 **return** $\boldsymbol{R}_t^*$;

recruit as high-quality workers as possible within a limited budget. Therefore, we designed the following strategy. Concretely, in each time slot t, we calculate the UCB-based quality $\hat{q}_i^t$ for each worker w_i, then sort the workers in descending order by $\hat{q}_i^t$. The platform selects the scheduling workers according to the order iteratively. Let $\mathcal{W}_t^s$ denote the set of currently selected workers, $\mathcal{W}_t^{s'}$ denote the set of workers selected in the previous iteration, $\boldsymbol{R}_t^{s,*}$ denote the optimal resource purchase strategy obtained by Algorithm 2 based on $\mathcal{W}_t^s$. In each iteration, the platform selects the unselected workers with the highest $\hat{q}_i^t$ into $\mathcal{W}_t^s$ and uses Algorithm 2 to make resource purchase strategy $\boldsymbol{R}_t^{s,*}$. If the difference between $G(\boldsymbol{R}_t^{s,*})$ and $G(\boldsymbol{R}_t^{s',*})$ is less than ϵ, then stop the iteration and set $\boldsymbol{R}_t^*$ equal to $\boldsymbol{R}_t^{s,*}$; otherwise, continue iterating. This method incrementally optimizes the platform's utility, helping to avoid inefficient consumption of the platform's budget.

We illustrate the pseudo-code of the worker scheduling in Algorithm 3, the pseudo-code consists of worker scheduling with unknown quality and optimal resource purchase strategy formulation. We do some necessary initialization in line 1. Then, we make a judgment if the current is the initial time slot, if yes, then we recruit all workers in line 4 to explore all workers' qualities and set the quality of all workers to 1 in line 5. After that, we randomly assign a task to each worker in line 6. Then, we make the resource purchase strategy for the unassigned task by Algorithm 2 in line 7. R_{ij}^t obtained from line 4 to line 7 together form $\boldsymbol{R}_t^*$ in the initial time slot. If the current is not the initial time slot, then we first set $\mathcal{W}_t^{s'}$ to empty set in line 9, which denotes that $\mathcal{W}_t^s$ has not been updated. $\boldsymbol{R}_t^{s',*}$ denote the corresponding optimal resource purchase strategy. In lines 10-13, we calculate the UCB-based quality $\hat{q}_i^t$. Then, we rank the workers in descending order of $\hat{q}_i^t$ in line 14. Next, we select the worker with the highest UCB-based quality into $\mathcal{W}_t^s$ and call Algorithm 2 to calculate the optimal resource purchase strategy based on $\mathcal{W}_t^s$ in line 16. After that, in lines 17-20, we determine whether the absolute value of the difference between $G(\boldsymbol{R}_t^{s,*})$ and $G(\boldsymbol{R}_t^{s',*})$ is greater than the error ϵ or not. If it is, then we continue the update; otherwise, we stop the update and return the final optimal purchase strategy $\boldsymbol{R}_t^*$. We use Algorithm 3 to solve the problem in line 3 of Algorithm 1.

In the initial exploration phase, the time complexity is at most $O(I_c|\mathcal{F}_t|)$. In the exploration phase, let $\mathcal{F}_t^s$ denote the collection of all feasible solutions based on $\mathcal{W}_t^s$, the time complexity of sorting is $O(m\log(m))$, and the time complexity of computing $G(\boldsymbol{R}_t^{s,*})$ is $O(I_c|\mathcal{F}_t^s|)$. This process is performed $|\mathcal{W}_t^s|$ times. Therefore, the time complexity of Algorithm 3 is $O\left(I_c|\mathcal{F}_t| + I_c|\mathcal{W}_t^s|\cdot|\mathcal{F}_t^s|\right)$.

5 Performance Analysis

In this section, we first analyze the convergence and approximation properties of Algorithm 1. Then, we analyze the regret upper bound of Algorithm 3.

5.1 Convergence and Approximation Analysis

It's easy to know that U_t is a bounded function due to constraints, for clarity, we let U_{min} and U_{max} be the upper and lower bounds of U_t on all time slots, respectively.

Theorem 5.1. *The solution obtained by Algorithm 1 meets constraint Equations (12) and (13).*

Proof. Based on Lemma 4.6 in Reference [22], we have $\Delta(\Theta(t)) \leq B + \Omega(t)$, where B is a positive constant value, thus we can get the following inequation:

$$V \cdot U_{max} - \Delta(\Theta(t)) \geq V \cdot U_{min} - B - \Omega(t). \tag{22}$$

As $\Delta(\Theta(t)) \triangleq \mathbb{E}[L(\Theta(t+1)) - L(\Theta(t))|\Theta(t)]$ and $L(\Theta(t)) \triangleq \frac{1}{2}\sum_{i=1}^m Q_i^R(t)^2 + \frac{1}{2}Q^C(t)^2$, then, taking the summation of both sides of Equation (22) on $\mathcal{T}$, and combining with the Cauchy-Bunyakovsky-Schwarz inequality, we obtain

$$\left(\sum\nolimits_{w_i\in\mathcal{W}} Q_i^R(T) + Q^C(T)\right)^2 \leq 2T\left(B + V(U_{max} - V_{min})\right) + 2\sum\nolimits_{t=1}^T \Omega(t). \tag{23}$$

Then, dividing both sides of Equation (23) by T^2 and taking the square root of it, we have

$$\frac{\sum_{w_i\in\mathcal{W}} Q_i^R(T) + Q^C(T)}{T} \leq \sqrt{\frac{2\left(B + V(U_{max} - V_{min})\right)}{T} + \frac{2\sum_{t=1}^T \Omega(t)}{T^2}}. \tag{24}$$

As is proved in Theorem 4.8 in Reference [22], all queues are mean rate stable, thus $\lim_{T\longrightarrow\infty}\frac{1}{T}\sum_{t=1}^T \Omega(t)$ has a constant upper bound. Then, taking expectations on both sides of Equation (24) and letting T tend to infinity, we can obtain

$$\lim_{T\longrightarrow\infty}\frac{\mathbb{E}\{\left(\sum_{w_i\in\mathcal{W}} Q_i^R(T) + Q^C(T)\right)\}}{T} \leq 0. \tag{25}$$

And because of $Q_i^R(T) \geq 0, \forall w_i \in \mathcal{W}, Q^C(T) \geq 0$, we have $\lim\limits_{T\longrightarrow\infty}\mathbb{E}\{Q_i^R(T)\}/T = 0, \forall w_i \in \mathcal{W}$ and $\lim\limits_{T\longrightarrow\infty}\mathbb{E}\{Q^C(T)\}/T = 0$. □

Let $\boldsymbol{R}_t^*$ be the optimal strategy for $\boldsymbol{P1}$ for time slot t, and $\boldsymbol{R}_t^p$ denote the strategy determined by Algorithm 1 and 2 in time slot t. Then, we have the following theorem.

THEOREM 5.2. *For any $\delta > 0$ and positive control parameter $V \geq 0$, we have*

$$U_{OPT} - \lim_{T \longrightarrow \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\{U_t(\boldsymbol{R}_t^p)\} \leq \frac{B'}{V} - \delta, \tag{26}$$

where $U_{OPT} = \lim_{T \longrightarrow \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\{U_t(\boldsymbol{R}_t^*)\}$, and $B' = B + \frac{1}{\gamma} \log |\mathcal{F}_t|$. Recall that $\frac{1}{\gamma} \log |\mathcal{F}_t|$ is the Markov approximation optimal gap.

PROOF. Let us recall the model description in Section 2. Platform utility on each time slot is related to the process of task arrival. According to Theorem 4.5 in Reference [22], if the process of task arrival is stationary, then for any $\delta > 0$, we have

$$U_{OPT} \leq \mathbb{E}\{U_t(\boldsymbol{R}_t^p)\} + \delta, \tag{27}$$

$$\mathbb{E}\{E_i^R(t)\} \leq \delta, \forall i \in \mathcal{W}, \; \mathbb{E}\{E^C(t)\} \leq \delta. \tag{28}$$

Combining Lemma 4.6 in Reference [22], the following inequality can be obtained:

$$V \cdot \mathbb{E}\{U\left(\boldsymbol{R}_t^p\right)\} - \Delta\left(\Theta(t)\right) \geq V \cdot \mathbb{E}\{U\left(\boldsymbol{R}_t^p\right)\} - B' - \Omega(t) \geq V \cdot (U_{OPT} + \delta) - B'. \tag{29}$$

As $\Delta(\Theta(t)) \triangleq \mathbb{E}[L(\Theta(t+1)) - L(\Theta(t))|\Theta(t)]$, by summing the time slots t over $\mathcal{T}$ on both sides of Equation (29) and rearranging the terms, we have $U_{OPT} - \lim_{T \longrightarrow \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\{U_t(\boldsymbol{R}_t^p)\} \leq \frac{B'}{V} - \delta$. □

THEOREM 5.3. *For any positive control parameter $V \geq 0$, the time average expected virtual queue satisfies*

$$\limsup_{T \longrightarrow \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\{\|\Theta(t)\|_1\} \leq \frac{V \cdot (U_{max} - U_{min}) + B'}{\epsilon}. \tag{30}$$

PROOF. Suppose $\exists \epsilon \geq 0$ such that for all time slot $t \in \mathcal{T}$ and all possible values of $\Theta(t)$, according to Theorem 4.2 in Reference [22], we have

$$V \cdot \mathbb{E}\{U_t|\Theta(t)\} - \Delta\left(\Theta(t)\right) \geq V \cdot U_{min} - B' + \epsilon \left\|\Theta(t)\right\|_1. \tag{31}$$

Then, as $\Delta(\Theta(t)) \triangleq \mathbb{E}[L(\Theta(t+1)) - L(\Theta(t))|\Theta(t)]$ and U_{max} is the upper bound of U_t. Summing over $\mathcal{T}$ on both sides of Equation (31) and rearranging terms, we have

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\{\|\Theta(t)\|_1\} \leq \frac{V \cdot (U_{max} - U_{min}) + B'}{\epsilon} + C, \tag{32}$$

where $C = \frac{\mathbb{E}\{L(\Theta(0))\}}{\epsilon T}$. The theorem can be proved by setting $T \rightarrow \infty$ in Equation (32). □

Theorems 5.2 and 5.3 show that the gap between the utility obtained by MPLP-C and the optimal utility can be measured by $\mathcal{O}(1/V)$, and the size of the time average queue can be measured by $\mathcal{O}(V)$, which implies that we can adjust the control parameter V to achieve the balance between the optimal goal and the queue stability.

5.2 Regret Analysis

The main challenge of using UCB to solve the MAB problem lies in the balance between exploration and exploitation. In our problem, that is, the balance between exploring the quality of workers and scheduling workers with the highest quality who have been explored as many as possible. The general measurement metric of this balance is the regret, i.e., the quality loss due to not selecting the globally highest quality workers for each schedule in all time slots.

Before analyzing regret, we need to clarify a fact. During the scheduling of workers in time slot t, each time we select the worker with the highest quality, the size of the set of available workers decreases by 1. The size of the workers' set only affects the value of the quality regret's upper bound, not the structure of it. Therefore, we only analyze the quality regret in the situation where the platform selects a worker for the first time in every time slot. Similar conclusions can be drawn in the subsequent selection of workers.

Let q^* denote the highest expected quality among all workers, $S_i(t)$ denote the number of times worker w_i is selected as the first selection in each time slot up to time slot t, and I_t denote the first selected worker in time slot t. In addition, we denote the amount of I_t's resource purchased by the platform in time slot t as B_t^1. That is,

$$B_t^1 = \sum\nolimits_{a_j \in \mathcal{A}_t} R_{ij}^t, w_i = I_t, \forall t \in \mathcal{T}. \tag{33}$$

Then, the quality regret in the situation where the size of the worker set is $|\mathcal{W}| = m$ is defined as follows:

$$Re_m = \sum\nolimits_{i:q_i<q^*} \Delta_i \mathbb{E}\{S_i(T)\}, \tag{34}$$

where $\Delta_i = CB_i^T(q^* - q_i)$.

Theorem 5.4. *For $|\mathcal{W}| = m$, the upper bound of the expected quality regret after T time slots is as follows:*

$$Re_m \le \sum\nolimits_{i:q_i<q^*} \frac{8(CB_i^T)^2 \ln\left(\sum_{w_i \in \mathcal{W}} T b_i^{max}\right)}{\Delta_i} + \left(1 + \frac{\pi^2}{3}\right) \sum\nolimits_{i:q_i<q^*} \Delta_i. \tag{35}$$

Proof. In Algorithm 3, each worker is selected at least once in the initial time slot. Then,

$$S_i(t) = 1 + \sum\nolimits_{t=2}^{T} \mathbb{I}\{I_t = w_i\}. \tag{36}$$

By changing the original event $I_t = w_i$ to the union of two mutually exclusive events, we can rewrite Equation (36) as

$$S_i(t) = 1 + \sum\nolimits_{t=2}^{T} \mathbb{I}\{I_t = w_i, S_i(T-1) \ge k\} + \sum\nolimits_{t=2}^{T} \mathbb{I}\{I_t = w_i, S_i(T-1) < k\}. \tag{37}$$

Assuming that $\sum_{t=2}^{T} \mathbb{I}\{I_t = w_i, S_i(T-1) < k\} \ge k$, which means worker w_i is selected at least k times between time slot $t = 2$ and current time slot, i.e., $\exists t' \in \mathcal{T} \backslash t = 1, S_i(t') \ge k+1$. Then, we have $S_i(t'-1) \ge k$, which is contradictory to $S_i(T-1) < k$. Therefore, $\sum_{t=2}^{T} \mathbb{I}\{I_t = w_i, S_i(T-1) < k\} < k$, i.e., $\sum_{t=2}^{T} \mathbb{I}\{I_t = w_i, S_i(T-1) < k\} \le k-1$, put that into Equation (37), and we have

$$S_i(t) \le k + \sum\nolimits_{t=2}^{T} \mathbb{I}\{I_t = w_i, S_i(T-1) \ge k\}. \tag{38}$$

Let $\tilde{q}^{t,*}, \hat{q}^{t,*}$ and $\eta^{t,*}$ denote the highest learned quality, the highest UCB-based quality and the uncertainty measure associated with this UCB-based quality in time slot t, respectively. Combine the UCB-based quality $\hat{q}_i^t$, we can rewirte $\sum_{t=2}^{T} \mathbb{I}\{I_t = w_i, S_i(T-1) \ge k\}$ as $\sum_{t=2}^{T} \mathbb{I}\{\hat{q}^{t,*} \le \hat{q}_i^t, S_i(T-1) \ge k\}$. Due to:

$$\min_{0<s<t} \hat{q}^{s,*} \le \hat{q}^{t,*}, \hat{q}_i^t \le \max_{k \le s' < t} \hat{q}_i^{s'}, \tag{39}$$

We have the following inequation, and give further derivation:

$$\begin{aligned} S_i(t) &\le k + \sum\nolimits_{t=2}^{T} \mathbb{I}\{\min_{0<s<t} \hat{q}^{s,*} \le \max_{k \le s' < t} \hat{q}_i^{s'}\} \\ &\le k + \sum\nolimits_{t=2}^{T} \sum\nolimits_{s=1}^{t-1} \sum\nolimits_{s'=k}^{t-1} \mathbb{I}\{\hat{q}^{s,*} \le \hat{q}_i^{s'}\}. \end{aligned} \tag{40}$$

According to Equation (21), we can rewrite $\hat{q}^{s,*} \leq \hat{q}_i^{s'}$ as

$$\begin{aligned}\hat{q}^{s,*} - \hat{q}_i^{s'} &= \tilde{q}^{s,*} + \eta^{s,*} - (\tilde{q}_i^{s'} + \eta_i^{s'}) + q^* - q^* + q_i - q_i \\ &= (\tilde{q}^{s,*} + \eta^{s,*} - q^*) - (\tilde{q}_i^{s'} - \eta_i^{s'} - q_i) - (2\eta_i^{s'} + q_i - q^*) \leq 0.\end{aligned} \tag{41}$$

If inequality Equation (41) holds, then at least one of the following three events must hold.

(1) $\tilde{q}^{s,*} + \eta^{s,*} - q^* \leq 0 \Rightarrow \tilde{q}^{s,*} + \eta^{s,*} \leq q^*$. The maximum quality of workers based on UCB is lower than their actual quality. However, the UCB algorithm is based on the principle of optimism when facing uncertainty, i.e., $\hat{q}^{s,*}$ is greater than q^* generally. Therefore, the highest UCB-based quality of the workers is too low in this situation.
(2) $-(\tilde{q}_i^{s'} - \eta_i^{s'} - q_i) \leq 0 \Rightarrow \tilde{q}_i^{s'} \geq q_i + \eta_i^{s'}$. In this situation, the UCB-based quality of worker w_i is too high.
(3) $-(2\eta_i^{s'} + q_i - q^*) \leq 0 \Rightarrow q^* \leq q_i + 2\eta_i^{s'}$. This indicates that the highest UCB-based quality of the workers is smaller than the UCB-based quality of worker w_i, which is abnormal and usually does not occur.

In Equation (34), to calculate Re_m, we have to calculate $\mathbb{E}\{S_i(T)\}$ first. Based on Equation (40), we have

$$\begin{aligned}\mathbb{E}\{S_i(T)\} &\leq \mathbb{E}\left\{k + \sum_{t=2}^{T} \sum_{s=1}^{t-1} \sum_{s'=k}^{t-1} \mathbb{I}\{\hat{q}^{s,*} \leq \hat{q}_i^{s'}\}\right\} \\ &= k + \sum_{t=2}^{T} \sum_{s=1}^{t-1} \sum_{s'=k}^{t-1} \mathbb{P}\{\hat{q}^{s,*} \leq \hat{q}_i^{s'}\},\end{aligned} \tag{42}$$

where $\mathbb{P}\{\cdot\}$ represents the probability of the event occurring.

Based on our above analysis for inequality Equation (41), we can scale $\mathbb{P}\{\hat{q}^{s,*} \leq \hat{q}_i^{s'}\}$ as follows:

$$\mathbb{P}\{\hat{q}^{s,*} \leq \hat{q}_i^{s'}\} \leq \mathbb{P}\{\tilde{q}^{s,*} + \eta^{s,*} \leq q^*\} + \mathbb{P}\{\tilde{q}_i^{s'} \geq q_i + \eta_i^{s'}\} + \mathbb{P}\{q^* \leq q_i + 2\eta_i^{s'}\}. \tag{43}$$

Then, we use Chernoff-Hoeffding inequality to obtain the upper bound of $\mathbb{P}\{\tilde{q}^{s,*} + \eta^{s,*} \leq q^*\}$ and $\mathbb{P}\{\tilde{q}_i^{s'} \geq q_i + \eta_i^{s'}\}$:

$$\mathbb{P}\{\tilde{q}^{s,*} + \eta^{s,*} \leq q^*\} \leq \exp(-2(s\eta^{s,*})^2/s) = t^{-4}, \tag{44}$$

$$\mathbb{P}\{\tilde{q}_i^{s'} \geq q_i + \eta_i^{s'}\} \leq \exp(-2(s'\eta_i^{s'})^2/s') = t^{-4}. \tag{45}$$

For $\mathbb{P}\{q^* \leq q_i + 2\eta_i^{s'}\}$, as we do not want the corresponding event occurring, we can keep it always 0 by controlling the value of k. In other words, we will control the value of k so that $q^* > q_i + 2\eta_i^{s'}$ always holds. We reorganize $q^* > q_i + 2\eta_i^{s'}$ as follows:

$$\begin{aligned}& q^* - q_i + 2\eta_i^{s'} > 0 \\ \Rightarrow\ & CB_i^T(q^* - q_i) + 2CB_i^T\eta_i^{s'} > 0 \text{ (as } CB_i^T > 0) \\ \Rightarrow\ & \Delta_i - 2CB_i^T\eta_i^{s'} > 0,\end{aligned} \tag{46}$$

where $\eta_i^{s'} = \sqrt{\frac{2\ln(\sum_{w_i \in \mathcal{W}} CB_i^{s'-1})}{CB_i^{s'-1}}}$ according to Equation (20). Due to the resource budget b_i^{max} of the worker w_i in a single time slot, we have $CB_i^{s'-1} \leq Tb_i^{max}$, and as k is the minimum times that worker w_i is selected between slot $t = 2$ and current time slot, we have $CB_i^{s'-1} \geq k$. Thus, the following inequality holds:

$$\begin{aligned}& \Delta_i - 2CB_i^T\eta_i^{s'} \geq \Delta_i - 2CB_i^T\sqrt{\frac{2\ln\left(\sum_{w_i \in \mathcal{W}} Tb_i^{max}\right)}{k}} > 0 \\ \Rightarrow\ & k > \frac{8(CB_i^T)^2 \ln\left(\sum_{w_i \in \mathcal{W}} Tb_i^{max}\right)}{\Delta_i^2}.\end{aligned} \tag{47}$$

As k is an integer number, we can let $k = \lceil \frac{8(CB_i^T)^2 \ln(\sum_{w_i \in \mathcal{W}} Tb_i^{max})}{\Delta_i^2} \rceil$ to ensure $\mathbb{P}\{q^* \leq q_i + 2\eta_i^{s'}\} = 0$.

Therefore, by combining Equations (34), (42), (43), (44), (45), and (47), we can calculate the upper bound of the expected quality regret after T time slots:

$$\begin{aligned} Re_m &= \sum_{i:q_i<q^*} \Delta_i \mathbb{E}\{S_i(T)\} \\ &\leq \sum_{i:q_i<q^*} \Delta_i \left(k + \sum_{t=2}^{T} \sum_{s=1}^{t-1} \sum_{s'=k}^{t-1} \mathbb{P}\{\hat{q}^{s,*} \leq \hat{q}_i^{s'}\} \right) \\ &\leq \sum_{i:q_i<q^*} \Delta_i \left(\left\lceil \frac{8(CB_i^T)^2 \ln\left(\sum_{w_i \in \mathcal{W}} Tb_i^{max}\right)}{\Delta_i^2} \right\rceil + \sum_{t=2}^{T} \sum_{s=1}^{t-1} \sum_{s'=k}^{t-1} \left(\frac{1}{t^4} + \frac{1}{t^4} + 0 \right) \right) \\ &\leq \sum_{i:q_i<q^*} \Delta_i \left(\frac{8(CB_i^T)^2 \ln\left(\sum_{w_i \in \mathcal{W}} Tb_i^{max}\right)}{\Delta_i^2} + 1 + \sum_{t=1}^{\infty} \sum_{s=1}^{t} \sum_{s'=1}^{t} \left(\frac{1}{t^4} + \frac{1}{t^4} + 0 \right) \right) \\ &\leq \sum_{i:q_i<q^*} \Delta_i \left(\frac{8(CB_i^T)^2 \ln\left(\sum_{w_i \in \mathcal{W}} Tb_i^{max}\right)}{\Delta_i^2} + 1 + \sum_{t=1}^{\infty} \frac{2}{t^2} \right). \end{aligned} \tag{48}$$

According to Euler's solution to the Basel problem, i.e., $\sum_{t=1}^{\infty} \frac{1}{t^2} = \frac{\pi}{6}$, we have

$$Re_m \leq \sum\nolimits_{i:q_i<q^*} \frac{8(CB_i^T)^2 \ln\left(\sum_{w_i \in \mathcal{W}} Tb_i^{max}\right)}{\Delta_i} + \left(1 + \frac{\pi^2}{3}\right) \sum\nolimits_{i:q_i<q^*} \Delta_i. \tag{49}$$

Let B_{max} and q_{min} denote the maximum total resources and the minimum expected quality among all workers, respectively. Then, the upper bound of Δ_i is calculated as follows:

$$\Delta_i \leq B_{max}(q^* - q_{min}). \tag{50}$$

Further, for the upper bound of Re_m in Equation (49), we have

$$\begin{aligned} Re_m &\leq \sum\nolimits_{i:q_i<q^*} \frac{8(CB_i^T)^2 \ln\left(\sum_{w_i \in \mathcal{W}} Tb_i^{max}\right)}{\Delta_i} + \left(1 + \frac{\pi^2}{3}\right) \sum\nolimits_{i:q_i<q^*} \Delta_i \\ &\leq \frac{8mB_{max} \ln\left(\sum_{w_i \in \mathcal{W}} Tb_i^{max}\right)}{q^* - q_{min}} + \left(1 + \frac{\pi^2}{3}\right) mB_{max}(q^* - q_{min}) \\ &= O\left(\ln\left(\sum\nolimits_{w_i \in \mathcal{W}} Tb_i^{max}\right)\right), \end{aligned} \tag{51}$$

which demonstrates that our algorithm has a sublinear regret. □

6 Simulations

6.1 Experimental Settings

In the simulation experiment, we establish a total of $T = 1,500$ slots and assume the presence of 30 crowdsourcing workers. The long-term time-average resource constraint for each worker is randomly assigned within the range of [6.5, 16.5] per time slot, with a unit price of $\tau = 1$. The real quality of each worker is randomly assigned within the range of [0, 1]. The total budget for the platform to purchase worker resources is set to sixty to one hundred percent of the cost of the total resources of all workers. The number of tasks received by the platform in each time slot t follows a Poisson distribution with $\lambda = 2|\mathcal{W}|$. For task a_j, the minimum resource $r_{j,t}^{min}$ required to execute is a random number on [2, 5]. For the parameter in the profit function of task a_j, α_j

Table 3. Simulation Parameter Settings

Parameter	Values
Unit resource price τ of workers	1
Workers set size $\|\mathcal{W}\|$	30
Tasks set size $\|\mathcal{A}_t\|$ in time slot t	Poisson distribution ($\lambda = 2\|\mathcal{W}\|$)
Profit parameters α_j, β_j of a_j	[5, 15]
Total time slots T	1500
Long-term time-average resource constraint $\overline{B_i}$	[6.5, 16.5]
Resource budget b_i^{max} of w_i in a single time slot	[2, 6]
Task completion quality q_i^t of w_i in time slot t	[0, 1]
Minimal resource requirement $r_{j,t}^{min}$ of a_j in time slot t	[2, 5]

and β_j are randomly distributed in [5, 15]. b_i^{max} is a multiple of time-average resource, where the multiple is in the range of [2, 6]. We run the algorithm 100 times under each given setting, and the data points in our figures are the average results of 100 runs. Simulation parameter settings are summarized in Table 3.

We compare our proposed **Online Worker Scheduling (OWS)** algorithm (Algorithm 1, which integrates Algorithm 3) with three baseline algorithms to evaluate its performance:

— **OWS with Known Quality (OWS-KQ)**: This algorithm is based on our OWS algorithm, the difference is that the platform has perfect knowledge of the real quality of each worker. By eliminating the need for exploration, the platform directly schedules workers based on their actual qualities. This serves as an upper bound for our algorithm, as it operates under idealized conditions.

— **OWS Without Short-term constraints (OWS-WS)**: This algorithm is also based on our OWS algorithm, which ignores the short-term constraints of workers' resource provision Equation (5). The algorithm is used to evaluate the impact of short-term constraints on platform utility.

— **Greedy algorithm (Greedy)**: In this algorithm, we replace the long-term constraint with an average constraint on each time slot, i.e., we consider the worker's resource constraint and the platform's budget constraint to be the same on each time slot. Then, in each time slot, we use a greedy algorithm to schedule workers. The greedy criterion is to iteratively select the worker with the highest quality within the constraint range and match the worker with the task with the highest profit.

6.2 Preformance Evaluation

Impact of control parameter V: From Figure 2, we can see that the time-average platform utility increases as the control parameter V increases. The result matches the conclusion of Theorem 5.2, that is, the gap of the time-average platform utility between our algorithm and the optimal solution is limited by $O(1/V)$. In addition, overall, the time-average virtual queue size of the platform and workers also increases as V increases, and the result matches the conclusion of Theorem 5.3, that is, the size of the time-average virtual queue is limited by $O(V)$.

Role of virtual queues: As shown in Figure 3(a), when the platform consumes a large amount of budget to purchase worker resources in time slot t, the value of the budget virtual queue in time slot $t+1$ will increase. Due to the queue stability control of our algorithm, the budget consumption in time slot $t+1$ will be reduced accordingly. This correlation between the budget virtual queue and budget consumption is most evident during time slots 15 to 16 and 18 to 19. Notably, when the budget consumption does not exceed the long-term time-average budget, the virtual queue value

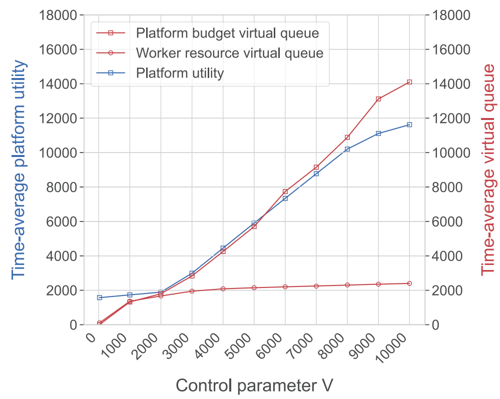
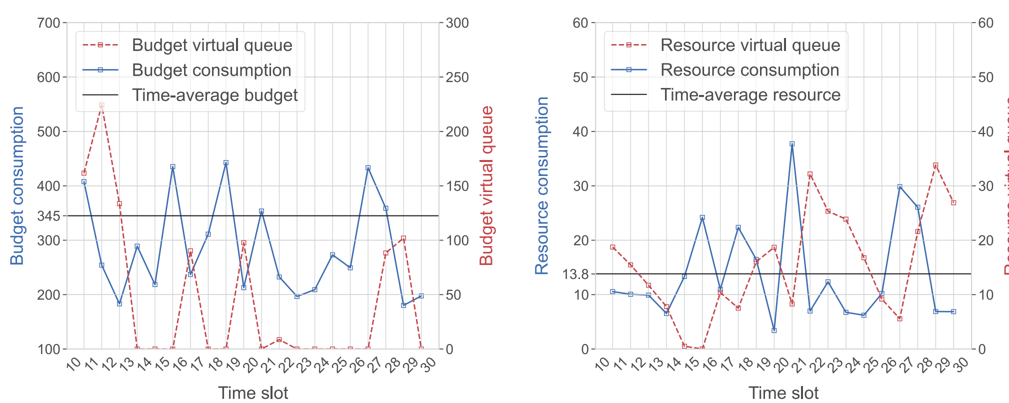


Fig. 2. Impact of control parameter V.

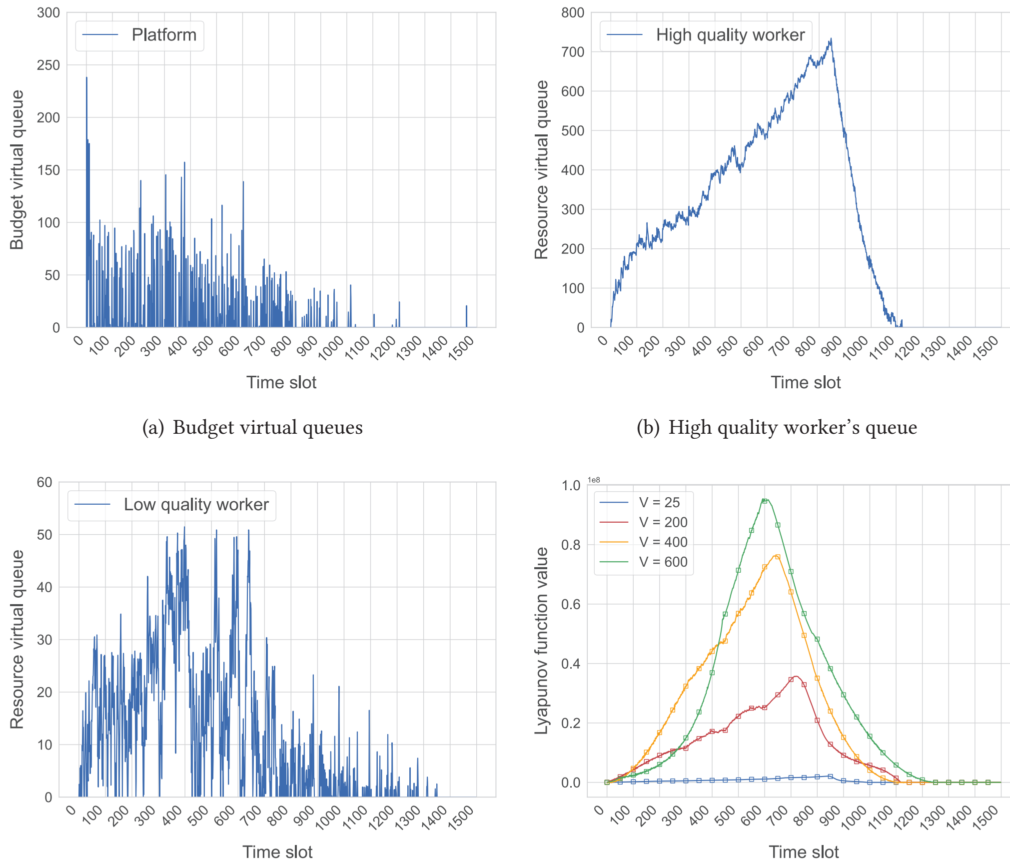


(a) Budget virtual queues (b) Resource virtual queues

Fig. 3. Analysis of relationship between virtual queue and consumption.

does not increase. This is determined by the definition of the virtual queue and can be clearly observed during time slots 21 to 27. In Figure 3(b), there is a similar relationship for worker resources between consumption and virtual queues. In time slots 21 to 25 in Figure 3(b), we can see that the worker resource consumption in multiple consecutive time slots does not exceed the time average resource amount. This is because too many resources are consumed in time slot 20, resulting in a large virtual queue. As a result, fewer resources are consumed in several consecutive time slots, thereby reducing the size of the virtual queue. This indirectly shows that our algorithm meets the long-term constraint of resource consumption by controlling the size of the virtual queue.

The queue stability: In our experiment, we set up 30 workers and one platform. Figures 4(a)–4(c), respectively, show the changes for the budget virtual queue of the platform, the resource virtual queue of the highest real quality worker, and the resource virtual queue of the lowest real quality worker over time, when $V = 25$. The virtual queue for workers with the highest real quality shows a clear trend of increasing and then converging, while the virtual queues for the platform and workers with the lowest real quality fluctuate and then decrease. This is because the OWS algorithm prioritizes selecting workers with high real quality to perform tasks in each time slot, resulting in workers with high true quality consuming higher resources in the first few time slots. However, due to the queue control mechanism of the algorithm, all queues eventually



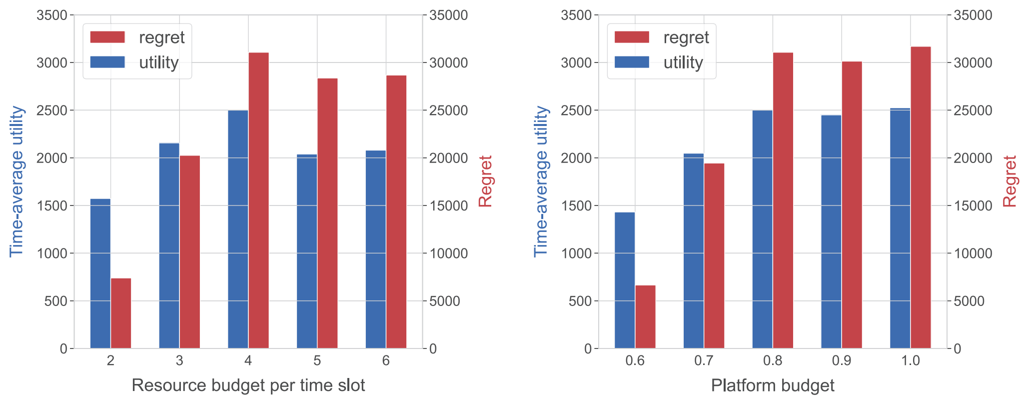
(a) Budget virtual queues (b) High quality worker's queue

(c) Low quality worker's queue (d) Lyapunov function value

Fig. 4. Analysis of the virtual queues stability.

converge in the later stages. According to the proof of Theorem 5.1, we can use the Lyapunov function (Equation (14)) to determine whether our algorithm satisfies the long-term constraints. As shown in Figure 4(d), We find that as the value of V increases, the time slot at which the Lyapunov function value reaches its peak occurs progressively earlier. This corresponds to the role of the control parameter V. In Section 4.2, we use V to balance the emphasis between platform utility and queue stability, with a larger V indicating a greater emphasis on platform utility. This means that as V increases, the platform consumes more worker resources in each time slot to maximize the objective function of Problem $P3$, resulting in the Lyapunov function value peaking earlier.

Utility and regret analysis: In our experimental setup, the resource budget b_i^{max} per worker per time slot is set as a multiple of the time-averaged resources, with the multiple ranging from [2, 6]. The platform budget C_{bgt} is set to sixty to one hundred percent of the cost of purchasing all workers' resources. In Figure 5, we compare the time-average utility and regret under different settings of b_i^{max} and C_{bgt} when $V = 25$. Figure 5(a) shows that with the increasing of b_i^{max}, time-average utility and regret also increase, but when the multiple reaches 4, the two reach a peak, and the further increase of b_i^{max} reduces it. This situation occurs because when b_i^{max} increases beyond a certain value, its limitation on the consumption of worker resources becomes weaker

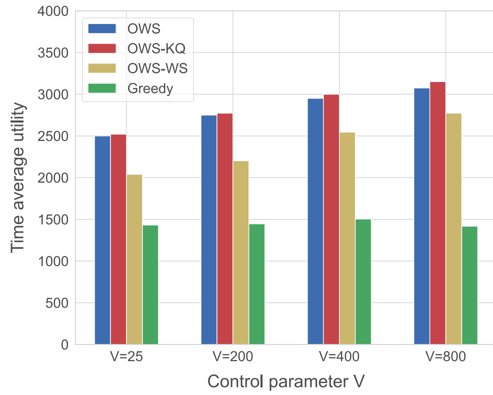


(a) Different b_i^{max} (b) Different C_{bgt}

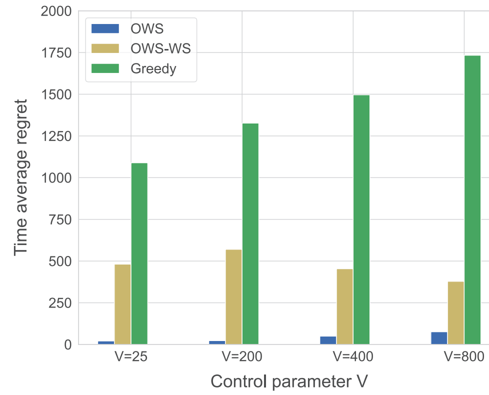
Fig. 5. Utility and regret analysis.

than the long-term constraint on worker resource consumption. In Figure 5(b), the time-average utility of the platform increases as C_{bgt} increases, but when it reaches eighty percent, the time-average utility stops growing. This is due to the limited long-term and short-term constraints on worker resources so that even if the platform has a sufficient budget, it can not consume more worker resources.

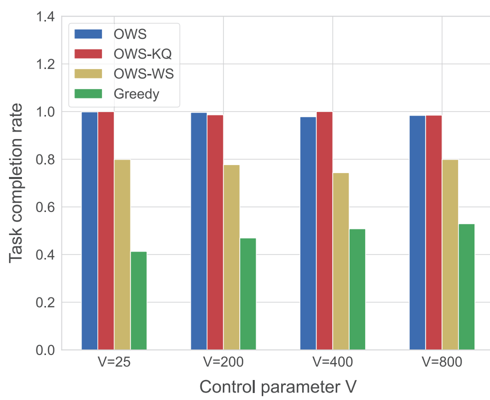
Algorithm comparison: In the most important aspect of the platform's utility, as shown in Figure 6(a), we compared the time-average utility of our proposed OWS algorithm and other baseline algorithms under different V conditions. Since we do not know the real quality of each worker at the beginning and use the UCB algorithm to learn the quality of workers, the time-average utility obtained by the OWS algorithm is slightly less than that obtained by the OWS-KQ algorithm with known worker's real quality. For the OWS-WS algorithm, while OWS-WS has fewer constraints compared to OWS, its performance is worse due to several reasons. First, by ignoring short-term constraints, OWS-WS may over-consume worker resources or budget in early time slots, leading to resource shortages in later time slots and reducing overall utility. Second, the absence of short-term constraints may result in the platform consuming more budget to purchase the resources of workers with low quality in the initial time slot, then reducing the utility. In contrast, OWS satisfies both short-term and long-term constraints, achieving better performance despite stricter constraints. The Greedy algorithm can only consume time average resources and budget at most on each time slot, resulting in its final task completion rate and resource utilization rate is low (as shown in Figures 6(c) and 6(d)), which makes its time-average utility low. In the aspect of regret, as shown in Figure 6(b), our algorithms are much smaller than algorithms OWS-WS and Greedy. Specifically, because the OWS-WS algorithm does not account for the short-term resource constraints of workers, the platform may purchase a significant amount of resources from workers with low expected quality in the initial time slots. This leads to an insufficient budget for purchasing resources from workers with high expected quality in the middle and later time slots, thereby increasing the algorithm's regret. For the Greedy algorithm, the platform lacks a sufficient budget to explore workers' expected quality during each recruitment, resulting in an extended exploration phase and inadequate exploration outcomes for recruiting workers with high expected quality, which leads to greater regret. Finally, in the aspect of task completion rate and resource utilization rate, as shown in Figures 6(c) and 6(d), The performance of our algorithm has a slight gap compared to the OWS-KQ algorithm, which is inevitable, because we need to explore



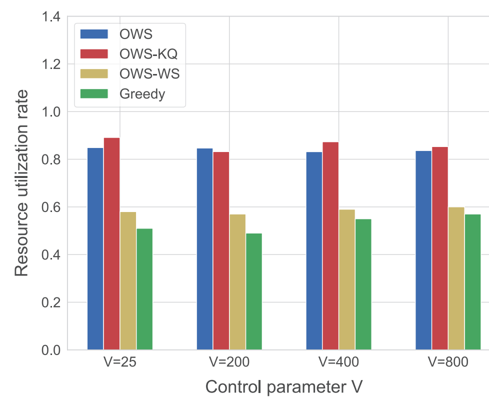
(a) Utility comparison



(b) Regret comparison



(c) Task completion comparison



(d) Resource utilization comparison

Fig. 6. Algorithm comparison.

the true quality of workers in initial time slots. However, compared to OWS-WS and Greedy, our algorithm achieves a higher task completion rate and resource utilization rate. For the OWS-WS algorithm, it initially consumed too much platform budget to purchase worker resources, resulting in insufficient budget in the later stages to acquire resources and execute tasks, which in turn led to lower task completion and resource utilization rates. For the Greedy algorithm, since the budget available for purchasing resources in each time slot is strictly limited, the amount of resources it can consume is lower than the total available worker resources, which in turn leads to a lower task completion rate and resource utilization rates.

7 Conclusion

In this article, we study the online worker scheduling problem with unknown quality for spatiotemporal crowdsourcing systems. The objective is to maximize the long-term utility of the platform under the long-term constraints of workers and the platform without knowing the real quality of the workers. To address the problem, we employ Lyapunov optimization techniques to decouple the long-term constraints and design an algorithm based on the UCB algorithm and Markov approximations to find solutions for each time slot. Extensive computer simulations have validated the efficacy and reliability of our designs.

References

[1] Stephen P. Boyd and Lieven Vandenberghe. 2004. *Convex Optimization.* Cambridge University Press, Cambridge, UK.

[2] Sha Chang, Su Deng, Yahui Wu et al. 2023. Online energy balancing strategy based on Lyapunov optimization in mobile crowdsensing. *IEEE Trans. Industr. Inform.* 19, 9 (Sep. 2023), 9266–9279.

[3] Chandra Chekuri and Sanjeev Khanna. 2005. A polynomial time approximation scheme for the multiple knapsack problem. *SIAM J. Comput.* 35, 3 (2005), 713–728.

[4] Minghua Chen. 2013. Markov approximation for combinatorial network optimization. *IEEE Trans. Inf. Theory* 59, 10 (2013), 6301–6327.

[5] Yu Ding, Lichen Zhang, and Longjiang Guo. 2022. Dynamic delayed-decision task assignment under spatial-temporal constraints in mobile crowdsensing. *IEEE Trans. Netw. Sci. Eng.* 9, 4 (2022), 2418–2431.

[6] Jiaang Duan, Jianfeng Lu, Wenchao Jiang et al. 2020. Incentivizing fairness-aware task allocation in mobile crowdsensing with sweep coverage and stability control. *Appl. Soft Comput.* 97 (2020), 106781.

[7] X. Fu, S. Long, A. Liu et al. 2024. Truthful and dual-direction combinatorial multi-armed bandit scheme to maximize profit for mobile crowd sensing. *IEEE Trans. Depend. Secure Comput.* (2024). DOI: 10.1109/TDSC.2024.3428405

[8] Guoju Gao, He Huang, Mingjun Xiao et al. 2022. Budgeted unknown worker recruitment for heterogeneous crowdsensing using CMAB. *IEEE Trans. Mobile Comput.* 21, 11 (Nov. 2022), 3895–3911.

[9] Hui Gao, Jianhao Feng, Yu Xiao et al. 2023. A UAV-assisted multi-task allocation method for mobile crowd sensing. *IEEE Trans. Mobile Comput.* 22, 7 (July 2023), 3790–3804.

[10] Shibo He, Kun Shi, Chen Liu et al. 2022. Collaborative sensing in Internet of Things: A comprehensive survey. *IEEE Commun. Surveys Tutor.* 24, 3 (Third Quarter 2022), 1435–1473.

[11] Sijie Huang, Guoju Gao, He Huang et al. 2024. Gather or scatter: Stackelberg-game-based task decision for blockchain-assisted socially aware crowdsensing framework. *IEEE Internet Things J.* 11, 2 (Jan. 2024), 1939–1951.

[12] Hongbo Jiang, Xingxia Dai, Zhu Xiao, and Arun Iyengar. 2023. Joint task offloading and resource allocation for energy-constrained mobile edge computing. *IEEE Trans. Mobile Comput.* 22, 7 (2023), 4000–4015.

[13] W. Jiang, P. Chen, W. Zhang et al. 2022. User recruitment algorithm for maximizing quality under limited budget in mobile crowdsensing. *Discrete Dynam. Nature Soc.* 2022, 1 (2022), 4804231.

[14] Feng Li, Yuqi Chai, Huan Yang et al. 2024. Incentivizing massive unknown workers for budget-limited crowdsensing: From off-line and on-line perspectives. *IEEE/ACM Trans. Netw.* 32, 3 (June 2024), 2406–2421.

[15] Jia Liu, Jianbo Shao, Min Sheng et al. 2024. Mobile crowdsensing ecosystem with combinatorial multi-armed bandit-based dynamic truth discovery. *IEEE Trans. Mobile Comput.* 23, 12 (2024), 13095–13113.

[16] Jia Liu, Jian Wang, and Guosheng Zhao. 2024. Conscious task recommendation via cognitive reasoning computing in mobile crowd sensing. *ACM Trans. Internet Technol.* 24, 4 (2024), 18:1–18:25.

[17] Xiaolong Liu, Honglong Chen, Yuping Liu et al. 2024. Multi-task data collection with limited budget in edge-assisted mobile crowdsensing. *IEEE Internet Things J.* 11, 9 (May 2024), 16845–16858.

[18] Zhuoran Lu, Yingjie Wang, Xiangrong Tong et al. 2023. Data-driven many-objective crowd worker selection for mobile crowdsourcing in industrial IoT. *IEEE Trans. Industr. Inf.* 19, 1 (Jan. 2023), 531–540.

[19] Mengtao Lyu, Xinyu Li, and Chun-Hsien Chen. 2022. Achieving knowledge-as-a-service in IIoT-driven smart manufacturing: A crowdsourcing-based continuous enrichment method for industrial knowledge graph. *Adv. Eng. Inform.* 51 (2022), 101494.

[20] Guoqi Ma, Honglong Chen, Yang Huang et al. 2023. Utility-based heterogeneous user recruitment of multitask in mobile crowdsensing. *IEEE Internet Things J.* 10, 11 (2023), 9796–9808.

[21] Wen Mo, Anfeng Liu, Neal N. Xiong et al. 2024. MPS: A truth discovery service scheme by using history data to maximize profit for mobile crowd sensing. *IEEE Trans. Serv. Comput.* (2024). DOI: 10.1109/TSC.2024.3433541

[22] Michael Neely. 2022. *Stochastic Network Optimization with Application to Communication and Queueing Systems.* Springer Nature.

[23] Tao Ouyang, Zhi Zhou, and Xu Chen. 2018. Follow me at the edge: Mobility-aware dynamic service placement for mobile edge computing. *IEEE J. Select. Areas Commun.* 36, 10 (2018), 2333–2345.

[24] Shuo Peng, Kun Liu, Shiji Wang et al. 2023. Time window-based online task assignment in mobile crowdsensing: Problems and algorithms. *Peer Peer Netw. Appl.* 16, 2 (2023), 1069–1087.

[25] Shuo Peng, Baoxian Zhang, Yan Yan et al. 2023. A multiplatform-cooperation-based task assignment mechanism for mobile crowdsensing. *IEEE Internet Things J.* 10, 19 (Oct. 2023), 16881–16894.

[26] Daniel Shevitz and Brad Paden. 1994. Lyapunov stability theory of nonsmooth systems. *IEEE Trans. Automat. Control* 39, 9 (1994), 1910–1914.

[27] Qinghua Sima, Guoju Gao, He Huang et al. 2022. Multi-armed bandits based task selection of a mobile crowdsensing worker. In *Proceedings of the International Conference on Computer Communications and Networks (ICCCN'22).* IEEE, 1–10.

[28] David Simchi-Levi and Chonghuan Wang. 2023. Multi-armed bandit experimental design: Online decision-making and adaptive inference. In *Proceedings of the International Conference on Artificial Intelligence and Statistics.* PMLR, 3086–3097.
[29] Shiwei Song, Zhidan Liu, Zhenjiang Li et al. 2020. Coverage-oriented task assignment for mobile crowdsensing. *IEEE Internet Things J.* 7, 8 (2020), 7407–7418.
[30] Jianheng Tang, Yishuo Cai, Saiqin Long et al. 2024. Q-BLPP: A quality-enabled bilateral location privacy-preserving service construction scheme in mobile crowd sensing. *IEEE Trans. Serv. Comput.* 14, 8 (Nov. 2024).
[31] Jianheng Tang, Feijiang Han, Kejia Fan et al. 2023. Credit and quality intelligent learning based multi-armed bandit scheme for unknown worker selection in multimedia MCS. *Inf. Sci.* 647 (2023), 119444.
[32] Qian Tao, Yongxin Tong, Zimu Zhou, Yexuan Shi, Lei Chen, and Ke Xu. 2020. Differentially private online task assignment in spatial crowdsourcing: A tree-based approach. In *Proceedings of the IEEE 36th International Conference on Data Engineering (ICDE'20).* IEEE, 517–528.
[33] Hengzhi Wang, Yongjian Yang, En Wang et al. 2023. Truthful user recruitment for cooperative crowdsensing task: A combinatorial multi-armed bandit approach. *IEEE Trans. Mobile Comput.* 22, 7 (July 2023), 4314–4331.
[34] Jiajun Wang, Xingjian Ding, Jianxiong Guo, Zhiqing Tang, and Deying Li. 2024. Online worker scheduling for maximizing long-term utility in spatiotemporal crowdsensing. In *Proceedings of the IEEE/CIC International Conference on Communications in China (ICCC'24).* IEEE, 1934–1939.
[35] Jian Wang, Delei Zhao, and Guosheng Zhao. 2024. Malicious participants and fake task detection incorporating gaussian bias. *ACM Trans. Internet Technol.* 24, 4 (2024), 19:1–19:19.
[36] Liang Wang, Dingqi Yang, Zhiwen Yu et al. 2023. Acceptance-aware mobile crowdsourcing worker recruitment in social networks. *IEEE Trans. Mobile Comput.* 22, 2 (Feb. 2023), 634–646.
[37] Zhenning Wang, Yue Cao, Kai Jiang et al. 2024. When crowdsensing meets smart cities: A comprehensive survey and new perspectives. *IEEE Commun. Surveys Tutor.* (2024).
[38] Zhibo Wang, Jing Zhao, Jiahui Hu et al. 2021. Towards personalized task-oriented worker recruitment in mobile crowdsensing. *IEEE Trans. Mobile Comput.* 20, 5 (May 2021), 2080–2093.
[39] Hui Xia, Ning Huang, Xuecai Feng et al. 2024. Starlet: Network defense resource allocation with multi-armed bandits for cloud-edge crowd sensing in IoT. *Digital Commun. Netw.* 10, 3 (2024), 586–596.
[40] Mingjun Xiao, Baoyi An, Jing Wang et al. 2022. CMAB-based reverse auction for unknown worker recruitment in mobile crowdsensing. *IEEE Trans. Mobile Comput.* 21, 10 (Oct. 2022), 3502–3518.
[41] Yunyi Xiao, Yu Yamashita, Hiroyoshi Ito et al. 2022. Multi-armed bandit approach to qualification task assignment across multi crowdsourcing platforms. In *Proceedings of the IEEE International Conference on Big Data (Big Data'22).* IEEE, 4040–4048.
[42] Chenghao Xu and Wei Song. 2023. Intelligent task allocation for mobile crowdsensing with graph attention network and deep reinforcement learning. *IEEE Trans. Netw. Sci. Eng.* 10, 2 (2023), 1032–1048.
[43] Yin Xu, Mingjun Xiao, Jie Wu, Sheng Zhang et al. 2022. Incentive mechanism for spatial crowdsourcing with unknown social-aware workers: A three-stage Stackelberg game approach. *IEEE Trans. Mobile Comput.* 22, 8 (2022), 4698–4713.
[44] Guisong Yang, Dongsheng Guo, Buye Wang et al. 2023. Participant-quantity-aware online task allocation in mobile crowd sensing. *IEEE Internet Things J.* 10, 24 (Dec. 15 2023), 22650–22663.
[45] Yanlin Zhang, Peng Li, Tao Zhang et al. 2023. Dynamic user recruitment in edge-aided mobile crowdsensing. *IEEE Trans. Vehic. Technol.* 72, 7 (July 2023), 9351–9365.
[46] Hui Zhao, Mingjun Xiao, and Jie Wu. 2021. Differentially private unknown worker recruitment for mobile crowdsensing using multi-armed bandits. *IEEE Trans. Mobile Comput.* 20, 9 (2021), 2779–2794.

Received 31 October 2024; revised 31 October 2024; accepted 3 December 2024