Differentially Private Unknown Worker Recruitment for Mobile Crowdsensing Using Multi-Armed Bandits

Hui Zhao, Mingjun Xiao, *Member, IEEE,* Jie Wu, *Fellow, IEEE,*
Yun Xu, *Member, IEEE,* He Huang, *Member, IEEE,* and Sheng Zhang, *Member, IEEE*

Abstract—Mobile crowdsensing is a new paradigm by which a platform can recruit mobile workers to perform some sensing tasks by using their smart mobile devices. In this paper, we focus on a privacy-preserving unknown worker recruitment issue. The platform needs to recruit some workers without knowing the qualities of them completing tasks. Meanwhile, these quality information also needs to be protected from disclosure. To tackle these challenges, we model the unknown worker recruitment as a Differentially Private Multi-Armed Bandit (DP-MAB) game by seeing each worker as an arm of DP-MAB and the task completion quality contributed by each worker as the reward of pulling arm. Then, recruiting workers is equivalent to designing a bandit policy of pulling DP-MAB arms. Under this model, we propose a Differentially Private ϵ-First-based arm-pulling (DPF) algorithm and a Differentially Private UCB-based arm-pulling (DPU) algorithm, which can achieve the nearly optimal expected accumulative rewards under a given budget. We also analyze the regrets of the DPF and DPU algorithms and prove that both of them are δ-differentially private on the task completion qualities ($\delta > 0$). Finally, we conduct extensive simulations to verify the significant performances of DPF and DPU based on both the real-trace and synthetic datasets.

Index Terms—Differential privacy, mobile crowdsensing, multi-armed bandit, worker recruitment

1 Introduction

With the explosive spread of smart mobile devices, Mobile CrowdSensing (MCS) has become an attractive paradigm for collecting sensing data. A typical MCS system consists of a platform residing on the cloud and a collection of mobile workers. The platform produces some sensing tasks and recruits mobile workers to perform these tasks by using their smart mobile devices. After completing the tasks, the workers will return the corresponding results to the platform. Since MCS can employ a lot of workers to complete a large task via their mobile devices, it has brought considerable flexibility to many applications, such as traffic information collection, noise pollution monitoring, indoor location, etc.

Worker recruitment is one of the most important issues in MCS systems. Existing worker recruitment mechanisms can be simply categorized into two models: the passive model and the proactive model. In the passive model, all tasks are publicized on the platform and workers directly apply for their preferred tasks to be performed. In the proactive model, the platform proactively recruits suitable workers to conduct the produced tasks. Since the platform in the proactive model can manipulate the worker recruitment process to optimize some metrics as it wants (e.g., to maximize the rewards, minimize the costs, etc.), this model attracts much research effort. Consequently, many worker recruitment algorithms have been proposed for various MCS systems [1]–[5]. At the same time, privacy-preserving issues and incentive mechanism design of worker recruitment have also been studied, e.g., [6]–[18].

In this paper, we focus on the issue of recruiting unknown workers for MCS systems. Although existing worker recruitment algorithms can deal with many MCS applications, most of them assume that the platform *knows* each worker's ability of performing diverse tasks, such as the successful probability of performing the task, the corresponding completion quality, and so on. Nevertheless, real MCS systems often do not support this assumption. Actually, it is difficult for a worker to evaluate its work skill and quality by itself in most cases. Thus, it is not realistic for the platform to know workers' Qualities of Completing tasks (QoCs) in advance. On the other hand, workers in most MCS systems are not familiar with each other, but they might compete for the same tasks. The workers are generally not willing to disclose their QoCs to others, since some sensitive private information might be revealed [7]. Therefore, for the unknown worker recruitment issue, we

- *H. Zhao, M. Xiao, and Y. Xu are with the School of Computer Science and Technology / Suzhou Institute for Advanced Study, University of Science and Technology of China, Hefei, P. R. China. M. Xiao is also with State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China.*
- *J. Wu is with the Center for Networked Computing, Temple University, 1805 N. Broad Street, Philadelphia, PA 19122.*
- *H. Huang is with the School of Computer Science and Technology, Soochow University, Suzhou 215006, P.R. China.*
- *S. Zhang is with State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China.*

This research was supported by the National Key R&D Program of China (Grant No. 2018AAA0101204), the National Natural Science Foundation of China (NSFC) (Grant No. 61872330, 61572457, 61379132, U1709217, 61572342), NSF grants CNS 1757533, CNS 1629746, CNS 1564128, CNS 1449860, CNS 1461932, CNS 1460971, IIP 1439672, the NSF of Jiangsu Province in China (Grant No. BK20191194, BK20131174, BK2009150), and Anhui Initiative in Quantum Information Technologies (Grant No. AHY150300). The corresponding authors are Mingjun Xiao: xiaomj@ustc.edu.cn and He Huang: huangh@suda.edu.cn

not only need to select appropriate workers without any prior knowledge on their QoCs, but also need to protect the privacy of each worker's sensitive information from being revealed to other workers.

A Toy Example. Assume that an MCS platform produces a task to collect some location-aware noise pollution data by using mobile devices. The task will last for a long period of time and thus is divided into many rounds. There are two workers (A and B) in the MCS system competing for the task. The objective is to achieve a better QoC performance. However, the platform does not know workers A and B before, so there is no prior knowledge about their QoCs. Thus, the platform needs to strategically select A or B to conduct each round of data collection task. During the process, the platform will repeatedly and randomly select A and B in order to learn their true QoC values and discover the better worker. Meanwhile, it will also leverage the knowledge (i.e., the QoCs of A and B) it has learnt to select the worker who is potentially the better worker for this task. That is to say, designing a worker recruitment strategy needs to take learning workers' QoC values into consideration. In addition, another worker C might pretend to participate in the task and eavesdrop the QoC values of other workers, which can reveal their sensitive information, such as their locations, professions, hobbies, etc. Besides, C can manipulate its own QoC and observe the corresponding sequential recruitment results published by the platform to infer the QoC ranges of other workers. Thus, protecting workers' QoCs from being revealed also needs be considered.

To tackle the above challenges, we treat the unknown worker recruitment of MCS as an online reinforcement learning process. On one hand, the platform repeatedly estimates workers' QoCs by recruiting them to perform some tasks, generally called the *exploration* process; on the other hand, based on the estimated QoCs, the platform continuously adjusts the recruitment policy to improve the total task completion quality, also known as the *exploitation* process. Since Multi-Armed Bandit (MAB) is an efficient reinforcement learning model to handle this kind of exploration versus exploitation dilemma [19], [20], we model our unknown worker recruitment problem as a Differentially Private MAB (DP-MAB) game, where each worker is seen as an MAB arm, recruiting a worker means pulling the corresponding arm, and the task completion quality contributed by the worker is seen as the reward of pulling the arm. Meanwhile, we treat the rewards of pulling arms (i.e., the recruited workers' QoCs) as a series of sensitive data, and adopt the differentially private mechanism to protect them from being revealed. The objective is to maximize the expected value of the accumulative reward (i.e., the total expected QoC), given a budget of worker recruitment cost.

So far, there have been substantial research on MAB. However, only a few works have investigated differentially private MAB problems [21], [22]. Moreover, none of them involves the costs and budgets of pulling arms. Different from these existing works, our DP-MAB model is derived from the unknown worker recruitment problem of MCS which takes into consideration the differential privacy of the rewards of pulling arms and the limited budget together. When introducing the costs and budget constraints into DP-MAB, our DP-MAB problem contains the 0-1 knapsack problems, which makes it more challengeable and completely different from the problems investigated in [21], [22]. To deal with this novel DP-MAB model, we extend the well-known ϵ-First and Upper Confidence Bound (UCB) bandit (a.k.a., arm-pulling) policies to propose a Differentially Private ϵ-First-based arm-pulling (DPF) algorithm and a Differentially Private UCB-based arm-pulling (DPU) algorithm, by which the platform can recruit suitable workers under a given budget. More specifically, our major contributions are summarized as follows:

1) We introduce a privacy-preserving unknown worker recruitment problem for MCS systems, where each worker's QoC follows an unknown distribution. We model it as a DP-MAB game with a limited budget, where recruiting unknown workers is turned to determining a bandit policy with the maximum expected accumulative reward. Unlike existing works, we consider the differential privacy of workers' QoCs and the recruitment budget simultaneously in our DP-MAB model.
2) We propose a budget-feasible ϵ-First differentially private bandit algorithm, i.e., DPF, by which the platform can recruit unknown workers to achieve a nearly optimal expected accumulative reward. Moreover, we analyze the corresponding online approximate performance to derive an upper bound on the regret (i.e., the expected reward loss). Also, we prove that the algorithm is δ-differentially private ($\delta > 0$).
3) We also propose a budget-feasible differentially private UCB-based bandit algorithm, i.e., DPU, for the platform recruiting unknown workers. Likewise, DPU can achieve the δ-differential privacy on workers' QoCs. Moreover, we also derive an upper bound on the regret of DPU.
4) We conduct extensive simulations on synthetic and real traces to evaluate the proposed DPF and DPU algorithms. Both of them demonstrate the significant performances. Moreover, when the budget of recruiting workers is small, DPF can obtain a better QoC performance than DPU; otherwise, if the budget becomes large, DPU will achieve a better QoC performance.

The remainder of the paper is organized as follows. We introduce the models and the problem in Section 2. The DPF and DPU algorithms are proposed in Sections 3 and 4, respectively. We evaluate their performances in Section 5. After reviewing the related works in Section 6, we discuss the possible extensions of our system in Section 7. The conclusion of this paper is presented in Section 8.

2 Models and Problem

2.1 System Overview

In this paper, we leverage an MCS system to continuously collect information for a period of time under a fixed monetary budget B, such as collecting daily noise pollution information in a month, or collecting real-time road-side parking availability information, etc. The system includes a platform and a collection of mobile workers who are willing

to perform the information collection task, denoted by a set $\mathcal{N} \stackrel{\text{def}}{=} \{1, \cdots, i, \cdots, N\}$. The information collection task is conducted periodically according to the following mode:

Definition 2.1 (**Periodical MCS Information Collection Mode**)**.** Time is divided into a series of equal-length time slots, denoted by $\mathcal{T} \stackrel{\text{def}}{=} \{1, \cdots, t, \cdots\}$. According to different realistic applications, a time slot might be one hour, one day, and so on. At the beginning of each time slot, the platform recruits a worker. Then, the worker performs the task and returns the corresponding results before the end of the time slot. The collected information will bring a reward to the MCS system. Hence, the platform will pay a certain monetary remuneration for the worker. This process will be continuously conducted until the given budget B is exhausted.

In the MCS system, workers' QoCs are unknown to the platform. We call them unknown workers:

Definition 2.2 (**Unknown Worker and QoC**)**.** For each worker $i \in \mathcal{N}$, we use a normalized nonnegative random variable $Q_{i,t} \in [0, 1]$ to denote its Quality of Completing the task (QoC) in an arbitrary time slot $t \in \mathcal{T}$. Moreover, $Q_{i,t}$ follows an unknown distribution with an unknown mean q_i, which is determined by the worker's ability. Since the distribution and mean are unknown, we call these workers unknown workers, or workers for short.

In addition, we also define the notations of cost and reward for each worker completing the task:

Definition 2.3 (**Cost, Reward, and Accumulative Reward**)**.** When worker $i \in \mathcal{N}$ is recruited by the platform to perform the task at time slot t, the worker will incur a *cost*, and the platform will produce a *reward*. The cost and the reward are denoted as c_i and $X_{i,t}$, respectively. In this paper, the reward contributed by each worker is actually the worker's QoC. Then, $X_{i,t} = Q_{i,t}$. When worker i is not recruited, we have $X_{i,t} = 0$. Moreover, we define $\vec{X}_t \stackrel{\text{def}}{=} (X_{1,t}, \cdots, X_{N,t})$ and let the sequence of the rewards contributed by worker i up to time slot t be denoted as $X_{i,1:t} = \{X_{i,1}, \cdots, X_{i,t}\}$. Additionally, the total reward contributed by worker i from time slot 1 to t is called the *accumulative reward*, denoted by $r_{i,t} = \sum_{j=1}^{t} X_{i,j}$.

Remark: Here, we assume that the reward contributed by a worker is equivalent to the QoC of this worker. This is reasonable since each worker will report the results of performing the task to the platform, and the platform can evaluate the QoC of the worker and represent it by using the reward contributed by the received results. Additionally, the objective of the platform is to recruit appropriate workers to maximize the *expected accumulative reward*. It is actually equivalent to maximizing the total expected QoC of all recruited workers.

2.2 Differentially Private Multi-Armed Bandit Model

MAB is a reinforcement learning model which is widely used to make a series of online decisions in an uncertain environment [19], [20]. A typical MAB model includes a slot machine with multiple *arms*. Each arm is associated with a *reward* drawn from an unknown distribution. A player will continuously pull the arms according to some strategy, called *bandit policy*, so as to maximize the expected accumulative reward.

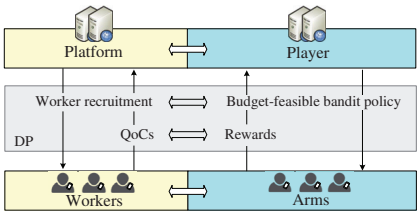


Fig. 1: The DP-MAB model

In our MCS system, the platform sequentially recruits unknown workers to perform the information collection task under the budget B, while protecting the privacy of the recruited workers. Taking the privacy and the budget into consideration, we model the unknown worker recruitment as a DP-MAB game, illustrated in Fig. 1. In the DP-MAB model, the platform is seen as a player, each worker in $\mathcal{N}$ is an arm, and the QoC of each recruited worker is seen as the reward of pulling the corresponding arm. In addition, the rewards of pulling arms are sensitive data to be protected via the differentially private mechanism. The objective of the platform is to sequentially pull the arms according to a budget-feasible bandit policy, so as to maximize the accumulative reward, while protecting the privacy of the rewards of pulling arms (i.e., workers' QoCs). Let $a_t \in \mathcal{N}$ denote the arm that the platform pulls in time slot t. Then, the bandit policy can be defined as follows:

Definition 2.4 (**Budget-Feasible Bandit Policy**)**.** A bandit policy Ψ is a sequence of maps: $\{\Psi_1, \cdots, \Psi_t, \cdots\}$, each of which specifies the arm that the platform will pull under the historical records, i.e., $a_t = \Psi_t(\vec{X}_{1:t-1})$, where $\vec{X}_{1:t-1} = (\vec{X}_1, \cdots, \vec{X}_{t-1})$. Moreover, the total cost is no larger than the given budget, i.e., $\sum_t c_{a_t} \leq B$.

While applying a bandit policy to pull the arms, we adopt the differential privacy mechanism to protect the values of reward of each arm in the whole process. The differential privacy can be formally defined as follows:

Definition 2.5 (**δ-Differential Privacy**([21], [23]))**.** A bandit policy Ψ is δ-differentially private if and only if over all time slots in $\mathcal{T}$, for all sequences $\vec{X}_{1:t-1}$ and $\vec{X}'_{1:t-1}$ differing in at most one time slot, and for any set $\mathcal{S} \subseteq \mathcal{N}$ we have

$$\mathbb{P}\{\Psi_t(\vec{X}_{1:t-1}) \in \mathcal{S}\} \leq e^{\delta} \cdot \mathbb{P}\{\Psi_t(\vec{X}'_{1:t-1}) \in \mathcal{S}\}. \quad (1)$$

Here, $\delta > 0$ is a small constant, indicating the privacy level that the policy provides.

Then, the accumulative reward $r_{i,t}$ which is manipulated by a differentially private mechanism is called the *disguised accumulative reward*. To make a distinction, we use $\hat{r}_{i,t}$ to denote the disguised accumulative reward.

Remark: Intuitively, for an arbitrary time slot t and a pair of reward sequences $\vec{X}_{1:t}$ and $\vec{X}'_{1:t}$ with at most one different reward vector, there at most exists one time slot $j \leq t$ such that $\vec{X}_j = (X_{1,j}, \cdots, X_{i,j}, \cdots, X_{N,j})$ is changed to $\vec{X}'_j = (X'_{1,j}, \cdots, X'_{i,j}, \cdots, X'_{N,j})$. Definition 2.5 means that if we change any reward vector $\vec{X}_j$ to $\vec{X}'_j$, the worker recruited by the bandit policy Ψ will not change too much at time slot $j+1$ or later on. This also indicates that an adversary will not be able to distinguish between the p-resence or absence of the reward vector $\vec{X}_j$ from the output

TABLE 1: Description of major notations

Variable	Description
$i, \mathcal{N}$	the i-th worker, and the set of all workers.
$\mathcal{T}$, $1:t$, t^+	the set of all time slots ($\mathcal{T} \stackrel{\text{def}}{=} \{1, \cdots, t, \cdots\}$), the time slots from 1 to t, and the time slots from t to the end time slot.
c_i, B, B_t	worker i's cost, the total budget of recruiting workers, and the residual budget at time slot t.
$Q_{i,t}$, q_i	worker i's Quality of Completing the task in time slot t, and the mean of $Q_{i,t}$ (Def. 2.2).
$X_{i,t}$, $X_{i,1:t}$, $\vec{X}_t$, $\vec{X}_{1:t-1}$	the reward contributed by worker i in time slot t, the sequence of these rewards until time slot t (Def. 2.3), $\vec{X}_t \stackrel{\text{def}}{=} (X_{1,t}, \cdots, X_{N,t})$, and $\vec{X}_{1:t-1} = (\vec{X}_1, \cdots, \vec{X}_{t-1})$.
$z_{i,t}$, z_{i,t^+}	the total number of times that worker i has been recruited from time slot 1 to t and from t to the end time slot (Sec. 2.3).
$r_{i,t}$, r	the accumulative reward contributed by worker i until time slot t (Def. 2.3), and the accumulative reward contributed by all workers until the budget expires.
$\hat{r}_{i,t}$	the disguised accumulative reward contributed by worker i until time slot t which computed by a hybrid mechanism.

(i.e., a recruited worker) of the differentially private bandit policy. In addition, malicious workers might eavesdrop the QoC values of others for the sake of acquiring their private information, and the QoC of a worker is equivalent to the reward of this worker which is contributed to the platform. Therefore, we mainly protect the differential privacy of the QoC sequence of each worker (i.e., the corresponding reward vector) from being revealed to other workers, except for the platform.

2.3 Problem Formalization

Under the DP-MAB model, the platform recruits the workers according to a bandit policy. The policy needs to i) satisfy δ-differential privacy over the whole recruitment process, as shown in Def. 2.5, ii) maximize the expected accumulative reward, and iii) guarantee that the total cost of pulling arms is no more than the given budget B. We use $z_{i,t}$ and z_{i,t^+} to denote the total number of times that the i-th arm has been pulled from time slot 1 to t and from t to the end time slot, respectively. Now, let r denote the accumulative reward that the platform obtains. Then, the expected accumulative reward $\mathbb{E}[r]$ can be calculated as follows:

$$\mathbb{E}[r] = \sum\nolimits_{i \in \mathcal{N}} q_i \mathbb{E}[z_{i,1^+}]. \quad (2)$$

And, the privacy-preserving unknown worker recruitment problem can be formulated as:

$$Maximize: \quad \mathbb{E}[r] \quad (3)$$

$$Subject\ to: \quad \sum\nolimits_{i \in \mathcal{N}} c_i z_{i,1^+} \leq B \quad (4)$$

$$\text{Eq. 1 holds.} \quad (5)$$

For ease of reference, we list the main notations in Table 1.

3 The DPF Bandit Algorithm

In this section, we propose a budget-feasible differentially private ϵ-First-based bandit algorithm, i.e., DPF, to solve the unknown worker recruitment problem. First, we model the unknown worker recruitment as a series of arm-pulling operations for a DP-MAB game. Under this model, the DPF algorithm adopts a budget-feasible ϵ-First bandit policy to determine workers, where the ϵ ratio of total budget is invested for learning the workers' QoCs (i.e., exploration) and the residual budget is used to select the best worker (i.e., exploitation). Meanwhile, the DPF algorithm leverages the hybrid differentially private mechanism to protect the privacy of workers' QoCs during the whole recruitment process. In the following subsections, we elaborate the main technologies, present the detailed algorithm, and proceed with the performance analyses.

3.1 The Hybrid Differentially Private Mechanism

Under the DP-MAB model, the platform conducts a series of arm-pulling operations for worker recruitment. In each time slot, the platform will determine an arm to pull according to the accumulative reward of each arm. If an arm is pulled, the corresponding worker's QoC will be added to the accumulative reward of this arm; otherwise, the corresponding accumulative reward will remain unchanged, which is equivalent to being added by 0. During this process, we apply the hybrid differentially private mechanism to protect the workers' QoCs from being revealed [24]. When the platform updates the accumulative reward of each arm, this mechanism will generate a Laplace noise for each incremental value (even though the incremental value might be 0). More specifically, we consider an arbitrary worker $i \in \mathcal{N}$, the rewards contributed by whom are $X_{i,1^+} = \{X_{i,1}, X_{i,2}, \cdots, X_{i,t}, \cdots\}$. Based on the hybrid differentially private mechanism, we introduce a function $\mathcal{H}_i(\cdot)$, which maps a series of rewards to a disguised accumulative reward by adding Laplace noises. Let $\text{Lap}(\lambda)$ denote a Laplace distribution with mean zero and scale λ, where the probability density function is denoted by $f(x)|_{\text{Lap}(\lambda)} = \frac{1}{2\lambda} \exp(-\frac{|x|}{\lambda})$. Then, when inputting $X_{i,1:t} = \{X_{i,1}, \cdots, X_{i,t}\}$, $\mathcal{H}_i(\cdot)$ can be calculated as follows:

$$\mathcal{H}_i(X_{i,1:t}) = \sum_{j=1}^{t} X_{i,j} + \text{Lap}(\frac{2N}{\delta}) + (k-1)\text{Lap}(\frac{2N\lfloor \log t \rfloor}{\delta}). \quad (6)$$

Here, k is the number of 1's in the binary expression of t, and the k Laplace noises are added at the time slot t. Moreover, we let the *disguised accumulative reward* of pulling the i-th arm be

$$\hat{r}_{i,t} = \mathcal{H}_i(X_{i,1:t}). \quad (7)$$

Then, for each arm, the platform can compute the corresponding disguised accumulative reward. In this way, the true value of each $X_{i,t}$ is protected from being revealed.

3.2 The Budget-Feasible ϵ-First Bandit Policy

In the DP-MAB model, the whole arm-pulling (i.e., the unknown worker recruitment) process is divided into the exploration and exploitation phases. To deal with the exploration versus exploitation dilemma, we propose a budget-feasible ϵ-First bandit policy. First, the platform determines a real number ϵ from the open interval $(0, 1)$ according to its historical experience. Then, it divides the budget B into two parts: ϵB for the exploration phase and $(1-\epsilon)B$ for the exploitation phase.

In the exploration phase, the platform estimates the mean reward of each arm (i.e., the mean QoC of the corresponding worker) by recruiting the worker to perform the task. Since there is no prior knowledge on workers' QoCs, we let each arm be tested equally. Without loss of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.2990221, IEEE Transactions on Mobile Computing

generality, we assume that the costs of all arms in $\mathcal{N}$ satisfy $c_1 \leq \cdots \leq c_N$. Then, the platform pulls the arms in $\mathcal{N}$ one by one in the non-decreasing order of their costs until the budget ϵB runs out. Let τ be the end time slot of the exploration phase, which satisfies:

$$\tau = \arg\max_{t} \sum_{j=1}^{t} c_{a_j} \leq \epsilon B. \tag{8}$$

Let $z_{i,\tau}$ denote the total number of times that the i-th arm has been pulled in this phase and let $\hat{r}_{i,\tau}$ denote the corresponding disguised accumulative reward. Then, they satisfy:

$$z_{i,\tau} \geq \lfloor \frac{\epsilon B}{\sum_{i=1}^{N} c_i} \rfloor, \quad \hat{r}_{i,\tau} = \mathcal{H}_i(X_{i,1:\tau}). \tag{9}$$

Here, $\lfloor \cdot \rfloor$ is the floor function. Based on Eqs. 8-9, the platform can estimate the mean value of the worker i's QoC, denoted by $\hat{q}_i$, satisfying $\hat{q}_i = \hat{r}_{i,\tau}/z_{i,\tau}$.

In the exploitation phase, the platform conducts arm-pulling operations according to the means of workers' QoCs (i.e., the estimated rewards of arms) that are estimated in the exploration phase. In order to maximize the expected accumulative reward within the budget constraint, we model the arm-pulling in this phase as a knapsack problem to be solved, where each arm is an item, the estimated mean reward of pulling an arm is the value of the item, the cost of pulling the arm corresponds to the weight of the item, and the budget $(1-\epsilon)B$ is seen as the capacity of the knapsack. Let $z_{i,(\tau+1)^+}$ denote the total number of times that the i-th arm is pulled in the exploitation phase. Then, the problem can be formulated as follows:

$$maximize: \quad \sum_{i=1}^{N} \hat{q}_i z_{i,(\tau+1)^+} \tag{10}$$

$$subject\ to: \quad \sum_{i=1}^{N} c_i z_{i,(\tau+1)^+} \leq (1-\epsilon)B \tag{11}$$

Since this knapsack problem is a well-known NP-hard problem, we adopt a greedy strategy to solve it. First, the platform computes the value per weight for each item, i.e., $\frac{\hat{q}_i}{c_i}$, which is called the *density* of the i-th arm. Then, the platform sorts the arms in the non-decreasing order of their densities. Next, in each time slot, the platform continuously pulls the arms with the highest density values until the budget $(1-\epsilon)B$ is exhausted. Each arm is allowed to be repeatedly pulled.

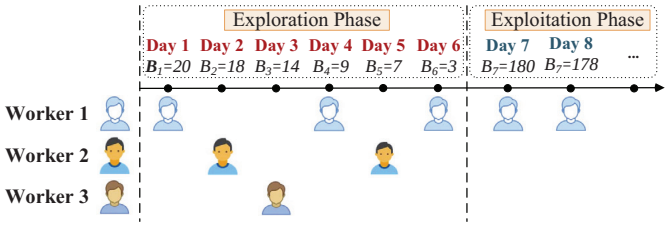
For better understanding, we follow the example in Section 1 to illustrate the budget-feasible ϵ-First bandit policy, as shown in Fig. 2. In this example, the monetary budget is 200, and three workers compete for the task, whose costs are $\{c_1 = 2, c_2 = 4, c_3 = 5\}$. Assume that the QoC of each worker follows the uniform distribution on [0,1], and the corresponding means are $q_1 = 0.4$, $q_2 = 0.6$, $q_3 = 0.8$, as shown in Figs. 2(a)-2(b). Let $\epsilon = 0.1$. Then, the budgets for exploration and exploitation are 20 and 180, respectively. Note that $c_1 < c_2 < c_3$. According to the budget-feasible ϵ-First bandit policy, in the exploration phase, the three workers will be recruited in the order of $\langle 1, 2, 3, 1, 2, 1 \rangle$ until the residual budget exhausts. At the end of exploration, we can compute the estimated QoC means for three workers: $\frac{r_{1,6}}{z_{1,6}} = \frac{0.6+0.3+0.5}{3} = \frac{1.4}{3}$, $\frac{r_{2,6}}{z_{2,6}} = \frac{0.7+0.5}{2} = 0.6$, $\frac{r_{3,6}}{z_{3,6}} = \frac{0.9}{1} = 0.9$. Accordingly, the estimated densities are $\frac{r_{1,6}}{c_1 z_{1,6}} = \frac{1.4}{6}$, $\frac{r_{2,6}}{c_2 z_{2,6}} = 0.15$, $\frac{r_{3,6}}{c_3 z_{3,6}} = 0.18$. Then, in the 7th day, the exploitation phase starts and the budget is 180, i.e., $B_7 = 180$. Since $\frac{r_{1,6}}{c_1 z_{1,6}} > \frac{r_{2,6}}{c_2 z_{2,6}} > \frac{r_{3,6}}{c_3 z_{3,6}}$, worker 1 will be always recruited until the residual budget becomes 0. Then, the whole recruitment process terminates and the recruitment order is $\langle 1, 2, 3, 1, 2, 1, 1, 1, \cdots, 1 \rangle$, as shown in Fig. 2(c). Here, for simplicity, we remove the Laplace noises produced by the hybrid differentially private mechanism.

	Exploration						Exploitation		
$X_{i,t}$	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	...
Worker 1	0.6	0.5	0.4	0.3	0.2	0.5	0.3	···	···
Worker 2	0.6	0.7	0.8	0.6	0.5	0.4	0.6	···	···
Worker 3	0.7	0.6	0.9	0.7	0.9	0.9	0.9	···	···

(a) The actual QoCs of each worker in different days

Cost c_i	Real QoC mean q_i	Estimated QoC mean $\frac{r_{i,6}}{z_{i,6}}$ obtained by exploration	density $\frac{r_{i,6}}{c_i z_{i,6}}$
2	0.4	1.4/3	1.4/6
4	0.6	0.6	0.15
5	0.8	0.9	0.18

(b) More information about the three workers



(c) The worker recruitment process

Fig. 2: An illustration of the budget-feasible ϵ-First bandit policy

3.3 The Detailed DPF Algorithm

The detailed DPF algorithm is shown in Algorithm 1. In Steps 3-8, we conduct the exploration process. More specifically, in Steps 5-6, we sequentially recruit the workers in $\mathcal{N}$. Whenever recruiting a worker, we judge whether the residual budget B_t is enough to recruit this worker. The exploration process terminates when the residual budget is less than the minimal cost c_1. In Step 8, we compute the disguised accumulative reward and the disguised estimated QoC mean of each worker. More specifically, if t can be represented as an integer power of 2, $\hat{r}_{i',t} = \hat{r}_{i',t-1} + \text{Lap}(2N/\delta)$; Otherwise, $\hat{r}_{i',t} = \hat{r}_{i',t-1} + (k-1)\text{Lap}(2N\lfloor \log t \rfloor/\delta)$. Here, k is the number of 1's in the binary expression of t. In Steps 12-20, we conduct the exploitation process, in which we recruit workers according to the greedy solution to the problem shown in Eqs. 12-13. That is, we greedily pull the arm with the highest value of $\hat{q}_i/c_i$ under the residual budget. In Step 18, we compute the disguised accumulative reward of each worker as in Step 8. The accumulative reward r over the whole process is computed in Step 21.

3.4 Performance Analysis

In this subsection, we prove the security, and analyze the regret performance and computation complexity of DPF.

Theorem 3.1. *The DPF algorithm is δ-differentially private.*

Proof: Consider an arbitrary time slot t and a pair of reward sequences $\vec{X}_{1:t}$ and $\vec{X}'_{1:t}$ with at most one different

Algorithm 1 The DPF Algorithm

Input: $\mathcal{N}$, $\{X_{i,t}, c_i | i \in \mathcal{N}, t \in \mathcal{T}\}$, B, ϵ, δ
Output: r

Initialization: $t=0$; $\forall i \in \mathcal{N}: z_{i,t}=0$;
Exploration phase:
$t=t+1$; Let $B_t=\epsilon B$ be the residual budget; Let $t'=t$;
while $B_t \geq c_1$ **do**
  while $i=t' \mod N$ **and** $B_t \geq c_i$ **do**
    $a_t=i$; Pull the a_t-th arm;
    $\forall i' \in \mathcal{N}$: $z_{i',t}=z_{i',t-1}$; $z_{a_t,t}=z_{a_t,t-1}+1$;
    $\forall i' \in \mathcal{N}$: $\hat{r}_{i',t}=\mathcal{H}_{i'}(X_{i',1:t})$; $\hat{q}_{i'}=\frac{\hat{r}_{i',t}}{z_{i',t}}$;
    $B_{t+1}=B_t-c_{a_t}$; $t=t+1$; $t'=t'+1$;
  $t'=1$;
Exploitation phase:
Let $B_t=(1-\epsilon)B$ be the residual budget;
Let $\mathcal{N}'=\mathcal{N}$ be the workers that have not been recruited;
while $B_t \geq c_1$ **do**
  while $a_t=\text{argmax}_{i \in \mathcal{N}'} \frac{\hat{q}_i}{c_i}$ **and** $B_t \geq c_{a_t}$ **do**
    Pull the a_t-th arm;
    $\forall i' \in \mathcal{N}: z_{i',t}=z_{i',t-1}$; $z_{a_t,t}=z_{a_t,t-1}+1$;
    $\forall i' \in \mathcal{N}$: $\hat{r}_{i',t}=\mathcal{H}_{i'}(X_{i',1:t})$;
    $B_{t+1}=B_t-c_{a_t}$; $t=t+1$;
  $\mathcal{N}'=\mathcal{N}'-\{a_t\}$;
$r=\sum_{i \in \mathcal{N}} \hat{r}_{i,t-1}$;

reward vector. That is, there at most exists one time slot $j \leq t$ such that $\vec{X}_j=(X_{1,j},\cdots,X_{i,j},\cdots,X_{N,j})$ is tampered to $\vec{X}'_j=(X'_{1,j},\cdots,X'_{i,j},\cdots,X'_{N,j})$. Then, for any worker i, $X_{i,1:t}$ and $X'_{i,1:t}$ differ in at most one reward record. Let $\Delta=\max_{j\in[1,t]}|X_{i,j}-X'_{i,j}|$. Since all rewards belong to [0,1], we have $\Delta \leq 1$, and $|\sum_{j=1}^t X_{i,j}-\sum_{j=1}^t X'_{i,j}| \leq \Delta \leq 1$. Then, for $r_i \in \mathbb{R}$, according to [23] and Eq. 6, we have:

$$\begin{aligned}
&\frac{\mathbb{P}\{\mathcal{H}_i(X_{i,1:t})=r_i\}}{\mathbb{P}\{\mathcal{H}_i(X'_{i,1:t})=r_i\}} \\
=&\frac{\mathbb{P}\{r_i-\sum_{j=1}^t X_{i,j}=\text{Lap}(\frac{2N}{\delta})+(k-1)\text{Lap}(\frac{2N\lfloor\log t\rfloor}{\delta})\}}{\mathbb{P}\{r_i-\sum_{j=1}^t X'_{i,j}=\text{Lap}(\frac{2N}{\delta})+(k-1)\text{Lap}(\frac{2N\lfloor\log t\rfloor}{\delta})\}} \\
=&\frac{f(r_i-\sum_{j=1}^t X_{i,j})|_{\text{Lap}(\frac{2N}{\delta})}\cdot[f(r_i-\sum_{j=1}^t X_{i,j})|_{\text{Lap}(\frac{2N\lfloor\log t\rfloor}{\delta})}]^{(k-1)}}{f(r_i-\sum_{j=1}^t X'_{i,j})|_{\text{Lap}(\frac{2N}{\delta})}\cdot[f(r_i-\sum_{j=1}^t X'_{i,j})|_{\text{Lap}(\frac{2N\lfloor\log t\rfloor}{\delta})}]^{(k-1)}} \\
\leq& e^{\frac{\delta}{2N}(1+\frac{k-1}{\lfloor\log t\rfloor})|\sum_{j=1}^t X_{i,j}-\sum_{j=1}^t X'_{i,j}|} \\
\leq& e^{\frac{\delta\Delta}{N}} \leq e^{\frac{\delta}{N}}
\end{aligned}$$

Here, k is the number of 1's in the binary expression of t. Thus, $k-1 \leq \lfloor\log t\rfloor$. Therefore, for each worker, the hybrid mechanism can guarantee that its reward sequence is $\frac{\delta}{N}$-differentially private. Now, we consider all workers. According to the composition property of differential privacy, for some $a \in \mathcal{N}$ we have:

$$\frac{\mathbb{P}\{\Psi(X_{1:t})=a\}}{\mathbb{P}\{\Psi(X'_{1:t})=a\}} \leq \frac{\prod_{i=1}^N \mathbb{P}\{\mathcal{H}_i(X_{i,1:t})=r_i\}}{\prod_{i=1}^N \mathbb{P}\{\mathcal{H}_i(X'_{i,1:t})=r_i\}} \leq e^{\delta}. \quad (12)$$

Therefore, we can conclude that the DPF algorithm is δ-differentially private. ■

Now, we derive an upper bound on *regret* of the DPF algorithm. Essentially, the regret is the expected loss of the reward achieved by DPF, compared to an optimal algorithm.

TABLE 2: Description of major formulas for DPF and DPU

Variable	Description
v_t	the upper bound of the sum of noises (Lemma 3.2).
i_*, $i_\diamond$	the arm with the maximal density and the arm the minimal density (Lemma 3.4).
σ, σ_i	the distance between the maximal density and the minimal density (Lemma 3.4), and the distance between the maximal density and the density of the i-th arm.
$I_{i,t}$	the UCB index of the i-th arm (Def. 4.1).
c_*, $c_\diamond$	the maximal cost and the minimal cost (Sec. 4.3).

Here, the optimal algorithm assumes that the platform has known the true QoC of each worker in advance and no privacy-preserving mechanisms are employed, so that it can make the optimal worker recruitment decision. Before the detailed theoretical analysis, we list the frequently used notations in Table 2 for clarity.

First, we introduce two lemmas which will be used in the derivation of regret bound:

Lemma 3.2 ([21], [24]). *Consider an arbitrary worker's accumulative reward $r_{i,t}$ ($=\sum_{j=1}^t X_{i,j}$) and the accumulative reward $\hat{r}_{i,t}$ disguised by using hybrid differentially private mechanism. Denote $v_t=\frac{\sqrt{8}}{\delta}\log(\frac{4}{\gamma})(\log t+1)$. Then, for any time slot $t \in \mathcal{T}$ and any $0<\gamma\leq t^{-b}$ ($b>0$), we have*

$$\mathbb{P}\{|\hat{r}_{i,t}-r_{i,t}| \geq v_t\} \leq \gamma.$$

Here, $|\hat{r}_{i,t}-r_{i,t}|$ equals to the sum of Laplace noises added to the accumulative reward $r_{i,t}$. And, v_t indicates an upper bound on the total Laplace noises with a high probability. According to [21], [24], we have $v_t=\frac{\sqrt{8}}{\delta}\log(\frac{4}{\gamma})(\log t+1)$.

Lemma 3.3 (Chernoff-Hoeffding bound). *Suppose that Y_1, Y_2, $\cdots$, Y_t are t random variables in the same range $[0,1]$, satisfying $\mathbb{E}[Y_j|Y_1,\cdots,Y_{j-1}]=\mu$ for $\forall j \in [1,t]$. Then, for any $\eta \geq 0$, we have:*

$$\mathbb{P}\{\sum_{j=1}^t Y_j \geq t\mu+\eta\} \leq e^{-\frac{2\eta^2}{t}},\ \mathbb{P}\{\sum_{j=1}^t Y_j \leq t\mu-\eta\} \leq e^{-\frac{2\eta^2}{t}}.$$

Based on the two lemmas, we have:

Lemma 3.4. *Denote the total accumulative reward produced by DPF at time slot t as $r_t=\sum_{i=1}^N r_{i,t}$, and denote the corresponding optimal total accumulative reward as r_t^*. Let τ ($\tau<t$) be the end time slot of the exploration phase and $v_t=\frac{\sqrt{8}}{\delta}\log(\frac{4}{\gamma})(\log t+1)$. Then, for any $\eta \geq 0$ and $0<\gamma\leq t^{-b}$ ($b>0$), with the probability at least $1-(e^{-\frac{2\eta^2}{t}}+\gamma)$, the expected regret $\mathbb{E}[r_t^*]-\mathbb{E}[r_t]$ satisfies:*

$$\mathbb{E}[r_t^*]-\mathbb{E}[r_t] \leq 2+\epsilon\sigma B+\frac{4(\eta+v_t)\sum_{i=1}^N c_i}{c_{i_*}}(\frac{1}{\epsilon}-1), \quad (13)$$

where $i_*=\underset{i\in N}{\text{argmax}}\ \frac{q_i}{c_i}$, $i_\diamond=\underset{i\in N}{\text{argmin}}\ \frac{q_i}{c_i}$, and $\sigma=\frac{q_{i_*}}{c_{i_*}}-\frac{q_{i_\diamond}}{c_{i_\diamond}}$.

Proof: First, according to the DPF algorithm, the total accumulative reward in the exploration phase satisfies:

$$\mathbb{E}[r_\tau] \geq \lfloor\epsilon B\frac{q_{i_\diamond}}{c_{i_\diamond}}\rfloor \geq \epsilon B\frac{q_{i_\diamond}}{c_{i_\diamond}}-1. \quad (14)$$

Second, we consider the exploitation phase. Note that at the end of the exploration phase, we have obtained the disguised estimated QoC $\hat{q}_i$ of each worker i, i.e., $\hat{q}_i=\frac{\hat{r}_{i,\tau}}{z_{i,\tau}}$. Based on this, we can capture the worker with the largest disguised QoC per cost. Let $\hat{i}_*$ be this worker, i.e., $\hat{i}_*=\text{argmax}_{i\in N}\frac{\hat{q}_i}{c_i}$. Then, according to the greedy arm-

pulling strategy in the exploitation phase, the expected accumulative reward produced by DPF, denoted by $r_{\tau:t}$, satisfies:

$$\mathbb{E}[r_{\tau:t}] \geq \lfloor \frac{(1-\epsilon)B}{c_{\hat{i}_*}} \rfloor q_{\hat{i}_*} \geq (1-\epsilon)B\frac{q_{\hat{i}_*}}{c_{\hat{i}_*}} - 1. \quad (15)$$

Next, we focus on the difference between the true value of an arbitrary QoC mean q_i and the corresponding disguised estimated value $\hat{q}_i$ produced by DPF. Since $r_{i,t} = \sum_{j=1}^{t} X_{i,j}$ denotes the sum of QoCs that worker i actually contributes to the platform, according to Lemma 3.2, we have:

$$\mathbb{P}\{|\hat{q}_i - \frac{r_{i,t}}{z_{i,t}}| \geq \frac{v_t}{z_{i,t}}\} = \mathbb{P}\{|\hat{r}_{i,t} - r_{i,t}| \geq v_t\} \leq \gamma. \quad (16)$$

At the same time, note that $z_{i,t}q_i$ denotes the expected value of accumulative reward $r_{i,t}$, i.e., $\mathbb{E}[r_{i,t}] = \mathbb{E}[\sum_{j=1}^{t} X_{i,j}] = z_{i,t}q_i$. Then, according to Lemma 3.3, we have:

$$\mathbb{P}\{|\frac{r_{i,t}}{z_{i,t}} - q_i| \geq \frac{\eta}{z_{i,t}}\} = \mathbb{P}\{|r_{i,t} - z_{i,t}q_i| \geq \eta\} \leq e^{-\frac{2\eta^2}{z_{i,t}}}. \quad (17)$$

Combining Eqs. 16-17, we can obtain:

$$\mathbb{P}\{|\hat{q}_i - q_i| \leq \frac{v_t + \eta}{z_{i,t}}\} \geq 1 - (e^{-\frac{2\eta^2}{z_{i,t}}} + \gamma). \quad (18)$$

Note that $t \geq z_{i,t} \geq z_{i,\tau} \geq \lfloor \frac{\epsilon B}{\sum_{i=1}^{N} c_i} \rfloor \geq \frac{\epsilon B}{2\sum_{i=1}^{N} c_i}$. Thus, we have:

$$\mathbb{P}\{|\hat{q}_i - q_i| \leq \frac{v_t + \eta}{\epsilon B/2\sum_{i=1}^{N} c_i}\} \geq 1 - (e^{-\frac{2\eta^2}{t}} + \gamma). \quad (19)$$

Finally, according to Eqs. 14-15, we have:

$$\begin{aligned}\mathbb{E}[r_t^*] - \mathbb{E}[r_t] &\leq B\frac{q_{i_*}}{c_{i_*}} - \mathbb{E}[r_\tau] - \mathbb{E}[r_{\tau:t}] \\ &\leq \epsilon\sigma B + (1-\epsilon)B(\frac{q_{i_*}}{c_{i_*}} - \frac{q_{\hat{i}_*}}{c_{\hat{i}_*}}) + 2. \quad (20)\end{aligned}$$

Further, according to Eq. 19, we can get:

$$\mathbb{P}\{|\hat{q}_{i_*} - q_{i_*}| \leq \frac{v_t + \eta}{\epsilon B/2\sum_{i=1}^{N} c_i}\} \geq 1 - (e^{-\frac{2\eta^2}{t}} + \gamma), \quad (21)$$

$$\mathbb{P}\{|\hat{q}_{\hat{i}_*} - q_{\hat{i}_*}| \leq \frac{v_t + \eta}{\epsilon B/2\sum_{i=1}^{N} c_i}\} \geq 1 - (e^{-\frac{2\eta^2}{t}} + \gamma). \quad (22)$$

Note that $\frac{\hat{q}_{\hat{i}_*}}{c_{\hat{i}_*}} \geq \frac{\hat{q}_{i_*}}{c_{i_*}}$. Then, combining Eqs. 20-22, we have

$$\begin{aligned}\mathbb{E}[r_t^*] - \mathbb{E}[r_t] &\leq 2 + \epsilon\sigma B + 2(1-\epsilon)B\frac{\eta + v_t}{c_{i_*} \cdot \epsilon B/2\sum_{i=1}^{N} c_i} \\ &\leq 2 + \epsilon\sigma B + \frac{4(\eta + v_t)\sum_{i=1}^{N} c_i}{c_{i_*}}(\frac{1}{\epsilon} - 1)\end{aligned}$$

with the probability of no less than $1-(e^{-\frac{2\eta^2}{t}} + \gamma)$. Therefore, the lemma holds. ■

Based on Lemma 3.4, we can set ϵ as a specific value which can minimize the upper bound in Eq. 13. Then, we have the following theorem:

Theorem 3.5. *When we set $\gamma = t^{-2}$ and $\epsilon = \left(\frac{4\sum_{i=1}^{N} c_i}{\sigma B c_{i_*}}\left(\eta + \frac{\sqrt{8}}{\delta}\log(\frac{2B}{c_1})(\log\frac{B}{c_1}+1)\right)\right)^{\frac{1}{2}}$, the upper bound on the regret shown in Eq. 13 can be tightened to $O(B^{\frac{1}{2}}\log(\frac{B}{c_1}))$.*

Proof: Let T be the end time slot when the DPF algorithm terminates. Then, $T \leq \frac{B}{c_1}$. Since $\gamma = t^{-2}$, we have $v_T \leq \frac{\sqrt{8}}{\delta}\log(2T)(\log T + 1) = \frac{\sqrt{8}}{\delta}\log(\frac{2B}{c_1})(\log\frac{B}{c_1} + 1)$. Then, according to Lemma 3.4, till the end time slot, we have

$$\begin{aligned}\mathbb{E}[r_T^*] - \mathbb{E}[r_T] &\leq 2 + \epsilon\sigma B \\ &+ \frac{4\sum_{i=1}^{N} c_i}{c_{i_*}}\left(\eta + \frac{\sqrt{8}}{\delta}\log(\frac{2B}{c_1})(\log\frac{B}{c_1}+1)\right)(\frac{1}{\epsilon} - 1).\end{aligned}$$

Finally, let $\epsilon = \left(\frac{4\sum_{i=1}^{N} c_i}{\sigma B c_{i_*}}\left(\eta + \frac{\sqrt{8}}{\delta}\log(\frac{2B}{c_1})(\log\frac{B}{c_1}+1)\right)\right)^{\frac{1}{2}}$, the above upper bound can achieve a minimal value:

$$\begin{aligned}&2 + 2\left(\frac{4\sigma B\sum_{i=1}^{N} c_i}{c_{i_*}}\left(\eta + \frac{\sqrt{8}}{\delta}\log(\frac{2B}{c_1})(\log\frac{B}{c_1}+1)\right)\right)^{\frac{1}{2}} \\ &- \frac{4\sum_{i=1}^{N} c_i}{c_{i_*}}\left(\eta + \frac{\sqrt{8}}{\delta}\log(\frac{2B}{c_1})(\log\frac{B}{c_1}+1)\right). \quad (23)\end{aligned}$$

Here, Eq. 23 is dominated by the second item, i.e., $2B^{\frac{1}{2}}\left(\frac{4\sigma\sum_{i=1}^{N} c_i}{c_{i_*}}\left(\eta + \frac{\sqrt{8}}{\delta}\log(\frac{2B}{c_1})(\log\frac{B}{c_1}+1)\right)\right)^{\frac{1}{2}}$, the order of which is $O(B^{\frac{1}{2}}\log(\frac{B}{c_1}))$. Therefore, the upper bound on the regret of DPU is tightened to $O(B^{\frac{1}{2}}\log(\frac{B}{c_1}))$. The theorem holds. ■

The above regret analysis shows that the reward loss of the DPF algorithm is mainly composed of two parts. One is due to the reason that the platform does not know the QoC of each worker. Another is incurred by the application of the differential privacy mechanism. When eliminating the effect of differential privacy from the above analysis, we can derive that the first part of reward loss will be bounded by $O(B^{\frac{2}{3}})$, which is aligned with the state-of-the-art results of similar MAB problems (e.g., [25], [26]). In other words, due to the application of the differential privacy mechanism, the regret bound of DPF changes from $O(B^{\frac{2}{3}})$ to $O(B^{\frac{1}{2}}\log(\frac{B}{c_1}))$. Despite this, it is still a sub-linear regret bound.

Theorem 3.6. *The DPF algorithm has a polynomial-time computational complexity.*

Proof: The computation overhead of Algorithm 1 is dominated by Steps 8, 15 and 18. In Steps 8 and 18, it needs to compute each worker's disguised accumulative reward by using the hybrid mechanism. According to [24], the corresponding computation overhead is $O(NB/c_1)$. Then, the computation overheads of Steps 8 and 18 are both $O(NB^2/c_1^2)$. In addition, in Step 15, we need to find the worker with the maximal density value, the overhead of which is $O(N\log N)$. Then, the computation complexity of Algorithm 1 is $O(N \cdot \max\{B^2/c_1^2, \log N\})$. Therefore, the theorem holds. ■

4 The DPU Bandit Algorithm

In this section, we propose a budget-feasible differentially private UCB bandit algorithm, i.e., DPU. In DPU, we extend the traditional UCB policy to be budget-feasible by taking the costs and budget of pulling arms into consideration simultaneously. In addition, the DPU algorithm also applies the same hybrid differentially private mechanism as in DPF to protect the privacy of workers' QoCs. Compared with the DPF algorithm in Section 3, DPU utilizes not just the estimated average workers' QoCs during the arm-pulling process, but also the upper confidence bounds of the estimated average QoCs. Moreover, DPU is more applicable to the scenarios where the budget B is large. The budget-feasible UCB policy, the detailed DPU algorithm and the performance analyses are presented as follows.

4.1 The Budget-Feasible UCB Policy

The traditional UCB policy computes an UCB index for each arm, which is composed of the current average reward and an upper bound of the corresponding confidence (of using the current value to estimate the true reward) [20]. The arm

with the maximal UCB index value will be pulled in each time slot. In this paper, we take the privacy into consideration, and thus define a novel concept, called the *differentially private UCB index*. This UCB index not only includes the average reward and the confidence upper bound, but also contains a corresponding Laplace noise. Besides, we take the costs and budget of pulling arms into consideration. Instead of just selecting an arm with the maximal differentially private UCB index value, we select multiple best arms within the budget constraint. This is formalized as a series of knapsack problems to be solved.

First, we assume that the platform has calculated the current accumulative reward contributed by each worker $i \in \mathcal{N}$ and has utilized the hybrid differentially private mechanism to disguise the value to get $\hat{r}_{i,t}$. Based on the disguised accumulative reward, we can define the differentially private UCB index as follows.

Definition 4.1 (Differentially Private UCB Index of Arm). The *differentially private UCB index* of the i-th arm, denoted by $I_{i,t}$, indicates the disguised expected average reward (i.e., the estimated QoC) and the size of the corresponding confidence interval, satisfying:

$$I_{i,t} = \frac{\hat{r}_{i,t}}{z_{i,t}} + \sqrt{\frac{2\ln t}{z_{i,t}}} + \frac{v_t}{z_{i,t}}, \tag{24}$$

where $z_{i,t}$ is the total number of times that the i-th arm has been pulled until time slot t, $v_t = \frac{\sqrt{8}}{\delta}\ln\frac{4}{\gamma}(\log t + 1)$, and $\sqrt{\frac{2\ln t}{z_{i,t}}} + \frac{v_t}{z_{i,t}}$ is an upper bound of confidence for the disguised accumulative reward.

Next, the platform seeks for the optimal bandit policy under the remaining budget as a reference for determining the arm to be pulled in the current time slot. This is also modeled as a knapsack problem, where the residual budget is the capacity of the knapsack, each arm is an item, and the pulling cost is seen as the weight. Moreover, based on the idea of the budget-feasible UCB policy, the differentially private UCB index of each arm is seen as the value of the corresponding item. Denote the remaining budget at time slot t by B_t. Then, the problem is formulated as follows:

$$maximize: \quad \sum_{i=1}^{N} z_{i,t^+} I_{i,t-1} \tag{25}$$

$$subject\ to: \quad \sum_{i=1}^{N} c_i z_{i,t^+} \le B_t \tag{26}$$

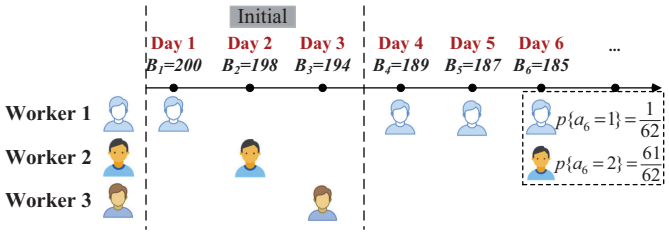
Finally, the platform solves the above problem to produce a solution $\{z_{1,t^+}, \cdots, z_{N,t^+}\}$ by using a greedy strategy. Based on the solution, the platform selects an arm $i \in \mathcal{N}$ to be pulled with the following probability:

$$\mathbb{P}\{a_t = i\} = \frac{z_{i,t^+}}{\sum_{i=1}^{N} z_{i,t^+}}. \tag{27}$$

In order to better understand the budget-feasible UCB bandit policy, we leverage the same example in Subsection 3.2 for illustration, as shown in Fig. 3. In the first three days, the MCS platform recruits the three workers one by one. Then, we can compute the UCB index of each worker as $I_{i,t} = \frac{r_{i,t}}{z_{i,t}} + \sqrt{\frac{2\ln t}{z_{i,t}}}$ and compute the density as $\frac{I_{i,t}}{c_i}$, i.e., $\frac{I_{1,3}}{c_1} = \frac{0.6+\sqrt{2\ln 3}}{2}$, $\frac{I_{2,3}}{c_2} = \frac{0.7+\sqrt{2\ln 3}}{4}$, $\frac{I_{3,3}}{c_3} = \frac{0.9+\sqrt{2\ln 3}}{5}$. Since $\frac{I_{1,3}}{c_1} > \frac{I_{2,3}}{c_2} > \frac{I_{3,3}}{c_3}$, the solution to the knapsack problem in Eqs. 25-26 will be $\{94, 0, 0\}$. Then, in the 4th day, worker 1 will be recruited with probability 1. Next, after observing the value of $X_{1,4}$, we recompute the UCB index densities of the three workers: $\frac{I_{1,4}}{c_1} = \frac{0.45+\sqrt{\ln 4}}{2}$, $\frac{I_{2,4}}{c_2} = \frac{0.7+\sqrt{2\ln 4}}{4}$, $\frac{I_{3,4}}{c_3} = \frac{0.9+\sqrt{2\ln 4}}{5}$. Since $\frac{I_{1,4}}{c_1} > \frac{I_{2,4}}{c_2} > \frac{I_{3,4}}{c_3}$, the solution to the knapsack problem will be $\{93,0,0\}$. Worker 1 will be still recruited with probability 1 in the 5th day, and the densities are updated as $\frac{I_{1,5}}{c_1} = \frac{(1.1/3)+\sqrt{(2\ln 5)/3}}{2}$, $\frac{I_{2,5}}{c_2} = \frac{0.7+\sqrt{2\ln 5}}{4}$, $\frac{I_{3,5}}{c_3} = \frac{0.9+\sqrt{2\ln 5}}{5}$. Since $\frac{I_{2,5}}{c_2} > \frac{I_{1,5}}{c_1} > \frac{I_{3,5}}{c_3}$, the solution to the knapsack problem will become $\{1, 61, 0\}$. Then, in the 6th day, worker 1 will be recruited with probability $\frac{1}{62}$, and worker 2 will be recruited with probability $\frac{61}{62}$. The above operations will terminate until the budget exhausts.

	Initial						
$X_{i,t}$	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	…
Worker 1	0.6	0.5	0.4	0.3	0.2	0.5	···
Worker 2	0.6	0.7	0.8	0.6	0.5	0.4	···
Worker 3	0.7	0.6	0.9	0.7	0.9	0.9	···

(a) The actual QoCs of each worker in different days



(b) The worker recruitment process

Fig. 3: An illustration of the budget-feasible UCB bandit policy

4.2 The Detailed DPU Algorithm

The detailed DPU algorithm is shown in Algorithm 2. In Step 3, we judge whether the residual budget B_t is enough for pulling an arm. When it is feasible, in Steps 4-6, we sequentially pull all arms in $\mathcal{N}$ once. Then, in Steps 8-9, we compute the value of $I_{i,t-1}$ for each arm. Based on this value, we solve the problem presented at Eqs. 25-26. That is, we greedily select the arm that has the largest value of $I_{i,t-1}/c_i$ under the residual budget. When obtaining the greedy solution $\{z_{1,t^+}, \cdots, z_{N,t^+}\}$, in Step 11, we select and pull an arm according to the probability distribution $\mathbb{P}\{a_t = i\} = \frac{z_{i,t^+}}{\sum_{i=1}^{N} z_{i,t^+}}$. When the budget is not enough to pull any arm, the algorithm terminates and releases the disguised accumulative reward. The disguised accumulative reward of each worker is computed in Step 13, which is the same as Steps 8 and 18 in Algorithm 1.

4.3 Performance Analysis

Since the DPU algorithm adopts the same hybrid differential private mechanism as DPF and the bandit policy cannot affect the privacy, we can directly get the following conclusion without any proof:

Theorem 4.2. *The DPU algorithm is δ-differentially private.*

Now, we analyze the regret performance of DPU by deriving an upper bound on the regret. To this end, we

Algorithm 2 The DPU Algorithm

Input: $\mathcal{N}$, $\{X_{i,t}, c_i | i\in\mathcal{N}, t\in\mathcal{T}\}$, B, δ, $c_\diamond = \min_i c_i$
Output: r

Initialization: $t=0$; $\forall i\in\mathcal{N}: z_{i,t}=0$;
$t=t+1$; Let $B_t=B$ be the residual budget;
while $B_t \geq c_\diamond$ **do**
  if $t\leq N$ **then**
    $a_t=t$; Pull the a_t-th arm;
    $\forall i\in\mathcal{N}$: $z_{i,t}=z_{i,t-1}$; $z_{a_t,t}=z_{a_t,t-1}+1$;
  else
    for each $i\in\mathcal{N}$ **do**
      Compute $I_{i,t-1}$ according to Def. 4.1;
    Solving the problem shown in Eqs. 25-26 to get $\{z_{1,t^+},\cdots,z_{N,t^+}\}$;
    Pull the a_t-th arm with probability shown in Eq. 27;
    $\forall i\in\mathcal{N}$: $z_{i,t}=z_{i,t-1}$; $z_{a_t,t}=z_{a_t,t-1}+1$;
  $\forall i\in\mathcal{N}$: $\hat{r}_{i,t}=\mathcal{H}_i(X_{i,1:t})$;
  $B_{t+1}=B_t-c_{a_t}$; $t=t+1$;
$r=\sum_{i\in\mathcal{N}}\hat{r}_{i,t-1}$;

assume that DPU terminates in time slot T. Under this assumption, we first calculate the probability of pulling an arbitrary arm (See Lemma 4.3). Then, with this probability, we analyze the expected value of the total number of times of pulling the arm (See Lemma 4.4). Next, we derive a lower bound on the total time slots of running DPU: T (See Lemma 4.5). Finally, based on the above lemmas, we can derive the expected regret produced by DPU.

For simplicity, we let $i_*=\text{argmax}_i\ \frac{q_i}{c_i}$, $\hat{i}_t=\text{argmax}_i\ \frac{I_{i,t-1}}{c_i}$, $c_*=\max_i c_i$, and $c_\diamond=\min_i c_i$. Then, we have:

Lemma 4.3. *Suppose that DPU terminates in time slot T. Then, for any $k\in\mathcal{N}$, and any $0<t\leq T$, we can get*

$$\mathbb{P}\{a_t=k|T\}\leq\mathbb{P}\{\hat{i}_t=k|T\}+(\frac{c_*}{c_\diamond})^2\frac{1}{T-t+1}. \tag{28}$$

Proof: For simplicity, we drop the conditional of T in this proof and will add it in the end. First, we consider a particular value of the residual budget B_t.

According to the greedy selection strategy, DPU will first select the $\hat{i}_t$-th arm at most $\lfloor\frac{B_t}{c_{\hat{i}_t}}\rfloor$ times. After this operation, the residual budget is at most $c_{\hat{i}_t}$. Then, we have $\sum_{i\neq\hat{i}_t} z_{i,t^+}\leq\frac{c_{\hat{i}_t}}{c_\diamond}$. If DPU only selects the arm with the largest cost, then $\sum_{i=1}^N z_{i,t^+}\geq\frac{B_t}{c_*}$. Therefore, combining these two inequalities, we have

$$\frac{\sum_{i\neq\hat{i}_t} z_{i,t^+}}{\sum_{i=1}^N z_{i,t^+}}\leq\frac{\frac{c_{\hat{i}_t}}{c_\diamond}}{\frac{B_t}{c_*}}\leq(\frac{c_*}{c_\diamond})^2\frac{c_\diamond}{B_t}.$$

Additionally, given the end time slot T, the DPU algorithm can still pull $T-t+1$ arms from time slot t, which means $B_t\geq c_{a_t}+c_{a_{t+1}}+\cdots+c_{a_T}\geq(T-t+1)c_\diamond$. Then, we can obtain $\frac{c_\diamond}{B_t}\leq\frac{1}{T-t+1}$. Further,

$$\frac{\sum_{i\neq\hat{i}_t} z_{i,t^+}}{\sum_{i=1}^N z_{i,t^+}}\leq(\frac{c_*}{c_\diamond})^2\frac{1}{T-t+1}.$$

Since DPU pulls the k-th arm with probability $\mathbb{P}\{a_t=k\}=\frac{z_{i,t^+}}{\sum_{i=1}^N z_{i,t^+}}$, we have

$$\begin{aligned}
&\mathbb{P}\{a_t=k|B_t\}\\
=&\mathbb{P}\{a_t=k,\hat{i}_t=k|B_t\}+\mathbb{P}\{a_t=k,\hat{i}_t\neq k|B_t\}\\
\leq&\frac{z_{\hat{i}_t}}{\sum_{i=1}^N z_{i,t^+}}\mathbb{P}\{\hat{i}_t=k|B_t\}+\frac{\sum_{i\neq\hat{i}_t} z_{i,t^+}}{\sum_{i=1}^N z_{i,t^+}}\mathbb{P}\{\hat{i}_t\neq k|B_t\}\\
\leq&\mathbb{P}\{\hat{i}_t=k|B_t\}+\frac{\sum_{i\neq\hat{i}_t} z_{i,t^+}}{\sum_{i=1}^N z_{i,t^+}}\\
\leq&\mathbb{P}\{\hat{i}_t=k|B_t\}+(\frac{c_*}{c_\diamond})^2\frac{1}{T-t+1}.
\end{aligned}$$

Next, for all possible values of the residual budget B_t, we have

$$\begin{aligned}
&\mathbb{P}\{a_t=k|T\}\leq\sum\nolimits_{B_t}\mathbb{P}\{a_t=k|T,B_t\}\mathbb{P}\{B_t|T\}\\
\leq&\sum\nolimits_{B_t}\big(\mathbb{P}\{\hat{i}_t=k|T,B_t\}+(\frac{c_*}{c_\diamond})^2\frac{1}{T-t+1}\big)\mathbb{P}\{B_t|T\}\\
\leq&\mathbb{P}\{\hat{i}_t=k|T\}+(\frac{c_*}{c_\diamond})^2\frac{1}{T-t+1}.
\end{aligned}$$

Therefore, the lemma holds. ■

Lemma 4.4. *Given that the DPU algorithm terminates in time slot T, for any $0<\delta,\rho<1$, we have*

$$\begin{aligned}
\mathbb{E}[z_{k,T}|T]\leq&1+\frac{2\pi^2}{3}+\max\{\alpha_k\ln T,\beta_k\ln(4T^4)(\log T+1)\}\\
&+(\frac{c_*}{c_\diamond})^2\ln T,
\end{aligned}\tag{29}$$

where $\alpha_k=\frac{8}{\rho^2\sigma_k^2c_k^2}$, $\beta_k=\frac{2\sqrt{8}}{\delta(1-\rho)\sigma_k c_{i_*}}$, $\sigma_k=\frac{q_{i_*}}{c_{i_*}}-\frac{q_k}{c_k}$.

Proof: We assume that T is given in advance. For simplicity, we drop the conditional of T in this proof and will add it in the end. According to Lemma 4.3, for any $l\geq 1$, we have

$$\begin{aligned}
&\mathbb{E}[z_{i,T}]=1+\sum\nolimits_{t=N+1}^T\mathbb{P}\{a_t=k\}\\
\leq&1+\sum\nolimits_{t=N+1}^T\mathbb{P}\{\hat{i}_t=k\}+\sum\nolimits_{t=N+1}^T(\frac{c_*}{c_\diamond})^2\frac{1}{T-t+1}\\
\leq&l+\sum\nolimits_{t=N+1}^T\mathbb{P}\{\hat{i}_t=k,z_{k,t}\geq l\}+\sum\nolimits_{t=N+1}^T(\frac{c_*}{c_\diamond})^2\frac{1}{T-t+1}.
\end{aligned}$$

For the second item in above equation, we have

$$\begin{aligned}
&\sum\nolimits_{t=N+1}^T\mathbb{P}\{\hat{i}_t=k,z_{k,t}\geq l\}\\
=&\sum\nolimits_{t=N+1}^T\mathbb{P}\{\frac{I_{i_*,t}}{c_{i_*}}\leq\frac{I_{k,t}}{c_k},z_{k,t}\geq l\}\\
\leq&\sum\nolimits_{t=N+1}^T\mathbb{P}\{\min_{1\leq z_{i_*,t}\leq t}\frac{I_{i_*,t}}{c_{i_*}}\leq\max_{l\leq z_{k,t}\leq t}\frac{I_{k,t}}{c_k}\}\\
\leq&\sum\nolimits_{t=1}^T\sum\nolimits_{z_{i_*,t}=1}^t\sum\nolimits_{z_{k,t}=l}^t\mathbb{P}\{\frac{I_{i_*,t}}{c_{i_*}}\leq\frac{I_{k,t}}{c_k}\}.
\end{aligned}$$

Note that $I_{i,t}=\frac{\hat{r}_{i,t}}{z_{i,t}}+\sqrt{\frac{2\ln t}{z_{i,t}}}+\frac{v_t}{z_{i,t}}$. Let $b_{t,n}=\sqrt{\frac{2\ln t}{n}}$. We can observe that if $\frac{I_{i_*,t}}{c_{i_*}}\leq\frac{I_{k,t}}{c_k}$ holds, then at least one of the following inequalities must hold:

$$\frac{I_{i_*,t}}{c_{i_*}}\leq\frac{q_{i_*}}{c_{i_*}};\quad\frac{I_{k,t}}{c_k}\geq\frac{q_k}{c_k};\quad\frac{q_{i_*}}{c_{i_*}}<\frac{q_k}{c_k}+\frac{2b_{t,z_{k,t}}}{c_k}+\frac{2v_t}{z_{i_*,t}c_{i_*}}. \tag{30}$$

For the first inequality in Eq. 30, we have

$$\begin{aligned}&\mathbb{P}\{\frac{I_{i_*,t}}{c_{i_*}}\le\frac{q_{i_*}}{c_{i_*}}\}\\&=\mathbb{P}\{\hat{r}_{i_*,t}\le r_{i_*,t}-v_t \text{ or } r_{i_*,t}\le z_{i_*,t}q_{i_*}-z_{i_*,t}b_{t,z_{i_*,t}}\}\\&\le\mathbb{P}\{\hat{r}_{i_*,t}\le r_{i_*,t}-v_t\}+\mathbb{P}\{r_{i_*,t}\le z_{i_*,t}q_{i_*}-z_{i_*,t}b_{t,z_{i_*,t}}\}.\end{aligned}$$

Here, according to Lemmas 3.2 and 3.3, we have $\mathbb{P}\{\hat{r}_{i_*,t}\le r_{i_*,t}-v_t\}\le\gamma$, and $\mathbb{P}\{r_{i_*,t}\le z_{i_*,t}q_{i_*}-z_{i_*,t}b_{t,z_{i_*,t}}\}\le e^{-2b^2_{t,z_{i_*,t}}z_{i_*,t}}=t^{-4}$. Thus, we can obtain $\mathbb{P}\{\frac{I_{i_*,t}}{c_{i_*}}\le\frac{q_{i_*}}{c_{i_*}}\}\le\gamma+t^{-4}$.

Similarly, for the second inequality in Eq. 30, we have $\mathbb{P}\{\frac{I_{k,t}}{c_k}\ge\frac{q_k}{c_k}\}\le\gamma+t^{-4}$. Let $\gamma=t^{-4}$. Then, we can conclude

$$\mathbb{P}\{\frac{I_{i_*,t}}{c_{i_*}}\le\frac{q_{i_*}}{c_{i_*}}\}\le2t^{-4},\text{ and }\mathbb{P}\{\frac{I_{k,t}}{c_k}\ge\frac{q_k}{c_k}\}\le2t^{-4}.\tag{31}$$

For the third inequality in Eq. 30, we need to find the minimum values of $z_{k,t}$ and $z_{i_*,t}$ such that it is false. Let $\sigma_i=\frac{q_{i_*}}{c_{i_*}}-\frac{q_i}{c_i}$. The event that the third inequality in Eq. 30 is false is equivalent to $\sigma_k\ge\frac{2b_{t,z_{k,t}}}{c_k}+\frac{2v_t}{z_{i_*,t}c_{i_*}}$. For any $0<\rho<1$, if the following two conditions hold:

$$\rho\sigma_k\ge\frac{2b_{t,z_{k,t}}}{c_k},\tag{32}$$

$$(1-\rho)\sigma_k\ge\frac{2v_t}{z_{i_*,t}c_{i_*}},\tag{33}$$

then $\sigma_k\ge\frac{2b_{t,z_{k,t}}}{c_k}+\frac{2v_t}{z_{i_*,t}c_{i_*}}$ is true. From the first condition in Eq. 32, we can obtain

$$z_{k,t}\ge\frac{8\ln t}{\rho^2\sigma_k^2c_k^2}.\tag{34}$$

From the second condition in Eq. 33, we have $z_{i_*,t}\ge\frac{2v_t}{(1-\rho)\sigma_kc_{i_*}}$. Since $v_t=\frac{\sqrt{8}}{\delta}\ln\frac{4}{\gamma}(\log t+1)$, and we have set $\gamma=t^{-4}$, we can obtain

$$z_{i_*,t}\ge\frac{2\sqrt{8}\ln(4t^4)}{\delta(1-\rho)\sigma_kc_{i_*}}(\log t+1).\tag{35}$$

Thus, under these two conditions Eqs. 34-35, the third inequality in Eq. 30 is false.

Let $\alpha_k=\frac{8}{\rho^2\sigma_k^2c_k^2}$, and $\beta_k=\frac{2\sqrt{8}}{\delta(1-\rho)\sigma_kc_{i_*}}$. We have

$$\begin{aligned}&l+\sum_{t=N+1}^{T}\mathbb{P}\{\hat{i}_t=k,z_{k,t}\ge l\}\\&\le\lceil\max\{\alpha_k\ln T,\beta_k\ln(4T^4)(\log T+1)\}\rceil+\sum_{t=1}^{T}\sum_{z_{i_*,t}=1}^{t}\sum_{z_{k,t}=l}^{t}4t^{-4}\\&\le1+\max\{\alpha_k\ln T,\beta_k\ln(4T^4)(\log T+1)\}+\frac{2\pi^2}{3}.\end{aligned}$$

Since $\sum_{t=N+1}^{T}(\frac{c_*}{c_\diamond})^2\frac{1}{T-t+1}\le(\frac{c_*}{c_\diamond})^2\ln T$, we can conclude

$$\begin{aligned}\mathbb{E}[z_{k,T}|T]\le1+\max\{\alpha_k\ln T,\beta_k\ln(4T^4)(\log T+1)\}+\frac{2\pi^2}{3}\\+(\frac{c_*}{c_\diamond})^2\ln T.\end{aligned}$$

Therefore, Lemma 4.4 holds. ■

Next, we derive an upper bound on the executive time slots T of DPU, which is shown as follows:

Lemma 4.5. *The total time slots of running the DPU algorithm, i.e., T, is bounded by*

$$\begin{aligned}\mathbb{E}[T]>\frac{B-c_\diamond}{c_{i_*}}-\sum_{k:c_k>c_{i_*}}\frac{c_k-c_{i_*}}{c_{i_*}}(1+\frac{2\pi^2}{3}+(\frac{c_*}{c_\diamond})^2\ln(\frac{B}{c_\diamond})\\+\max\{\alpha_k\ln(\frac{B}{c_\diamond}),\beta_k\ln(\frac{B}{c_\diamond})(\log(\frac{B}{c_\diamond})+1)\}).\end{aligned}$$

Proof: The DPU algorithm terminates if the residual budget is no more than the minimal cost, i.e., $B-\sum_{t=1}^{T}c_{a_t}<c_\diamond$. By using Lemma 4.4, we have

$$\begin{aligned}&B-c_\diamond<\mathbb{E}\Big[\sum_{t=1}^{T}c_{a_t}|T\Big]\\&\le\mathbb{E}\Big[\{\sum_{t=1}^{T}c_{i_*}+\sum_{k:c_k>c_{i_*}}(c_k-c_{i_*})\mathbb{P}\{a_t=k|T\}\}|T\Big]\\&\le\mathbb{E}[T]c_{i_*}+\sum_{k:c_k>c_{i_*}}(c_k-c_{i_*})\mathbb{E}\big[\mathbb{E}[z_{k,T}|T]\big]\\&\le\mathbb{E}[T]c_{i_*}+\sum_{k:c_k>c_{i_*}}(c_k-c_{i_*})\mathbb{E}\big[(1+\frac{2\pi^2}{3}+(\frac{c_*}{c_\diamond})^2\ln T\\&\qquad+\max\{\alpha_k\ln T,\beta_k\ln T(\log T+1)\})\big].\end{aligned}$$

Since $T\le B/c_\diamond$, by substituting $T=B/c_\diamond$ into the above equation we can directly prove that Lemma 4.5 holds. ■

Based on the above lemmas, we derive an upper bound on the regret of DPU, which is presented as follow:

Theorem 4.6. *For any budget $B>0$, the upper bound on the expected regret of the DPU algorithm is $O(\log^2(B/c_\diamond))$.*

Proof: The expected regret can be computed as follows:

$$\begin{aligned}&\mathbb{E}[r^*]-\sum_{k=1}^{N}q_k\mathbb{E}[z_{k,1^+}]\\&\le B\frac{q_{i_*}}{c_{i_*}}-q_{i_*}\sum_{k=1}^{N}\mathbb{E}[z_{k,1^+}]+\sum_{k=1}^{N}(q_{i_*}-q_k)\mathbb{E}\big[\mathbb{E}[z_{k,T}|T]\big]\\&\le q_{i_*}(\frac{B}{c_{i_*}}-\mathbb{E}[T])+\sum_{k:q_{i_*}>q_k}(q_{i_*}-q_k)\mathbb{E}\big[\mathbb{E}[z_{k,T}|T]\big]\\&\le(\frac{q_{i_*}}{c_{i_*}}\sum_{k:c_k>c_{i_*}}(c_k-c_{i_*})+\sum_{k:q_{i_*}>q_k}(q_{i_*}-q_k))(1+\frac{2\pi^2}{3}\\&+(\frac{c_*}{c_\diamond})^2\ln(\frac{B}{c_\diamond})+\max\{\alpha_k\ln(\frac{B}{c_\diamond}),\beta_k\ln(\frac{B}{c_\diamond})(\log(\frac{B}{c_\diamond})+1)\}).\end{aligned}$$

Here, Lemmas 4.4-4.5 are used in the last step. Moreover, the final result is in the order of $O(\log^2(B/c_\diamond))$. Therefore, the theorem holds. ■

Here, when ignoring the influence of the differential privacy mechanism in the above analysis, we can derive a regret bound $O(\log B)$, which corresponds to the reward loss only incurred by the platform not knowing the true QoC of each worker. It means that due to the application of the differential privacy mechanism, the regret bound of DPU increases from $O(\log B)$ to $O(\log^2(B/c_\diamond))$. This is still a sub-linear regret bound.

Note that we have obtained the upper bounds of regrets with regard to the DPF and DPU algorithms, which are shown in Theorems 3.5 and 4.6, respectively. By comparing the two regret bounds, we can infer that when the budget B is small, the regret of DPF will be smaller than that of DPU. However, when the budget B becomes larger, the regret of DPF will be more than that of DPU with a high probability. Therefore, we can conclude that when the budget B is small, the DPF algorithm can achieve a better QoC performance; otherwise, the DPU algorithm will be more suitable to obtain higher workers' QoCs. In Section 5, we will also verify this observation through sufficient simulations.

TABLE 3: Parameter Settings

Parameter name	Values
the budget of recruiting workers: B	1000 - 10000
the parameter in DPF: ϵ	0.01, 0.05, 0.1
the security parameter: δ	0.1 - 1.0

Finally, we consider the computational complexity of DPU.

Theorem 4.7. *The DPU algorithm has a polynomial-time computational complexity.*

Proof: The computation overhead of Algorithm 2 is dominated by Steps 10 and 13. In Step 10, we solve the knapsack problem, whose computation overhead is $O(N \log N \cdot B^2/c_\diamond^2)$. The computation overhead of Step 13 is the same as that of Steps 8 and 18 in Algorithm 1, which is $O(NB^2/c_\diamond^2)$. Then, the computation complexity of Algorithm 2 is $O(N \log N \cdot B^2/c_\diamond^2)$. Therefore, the theorem holds. ■

5 Evaluation

In this section, we conduct extensive simulations on real-trace and synthetic datasets to evaluate the performances of the DPF and DPU algorithms.

5.1 Evaluation Methodology

Algorithms for Comparison: In the simulations, we compare DPF and DPU with several representative algorithms, including ϵ_t-Greedy [19] and DP-UCB-Bound [21]. In each time slot t, ϵ_t-Greedy pulls an arm with the highest current estimated average reward with probability $1-\epsilon_t$ and selects a random arm with probability ϵ_t. Here, $\epsilon_t = \min\{1, \frac{5N}{t(q_{i*}-q_{i_\diamond})^2}\}$. The DP-UCB-Bound algorithm pulls the arm with the maximal value of $\frac{\hat{r}_{i,t}}{z_{i,t}} + \frac{4\sqrt{8}\log t(\log_2 z_{i,t}+1)}{\delta z_{i,t}}$. Since ϵ_t-Greedy can not guarantee differential privacy, for fair comparison, we incorporate the hybrid differentially private mechanism into ϵ_t-Greedy, i.e., using the disguised average reward as the estimated average reward. Moreover, since both of ϵ_t-Greedy and DP-UCB-Bound do not take the costs and budget of pulling arms into consideration, we add a cost to each arm consistently and conduct these algorithms under the same budget. In addition, we also implement the *optimal* (OPT) algorithm without privacy preservation for comparison. The OPT algorithm has full knowledge of the QoC value of each worker and recruits the optimal worker in each time slot.

Simulation Setup: We conduct our proposed algorithms and the compared algorithms using both the real-trace dataset and the synthetic datasets. The real-trace dataset we applied is Chicago Taxi Trips [27]. Due to the data reporting process, not all trips are reported and not all reported trips are usable. Here, we use the relatively complete trace reported in Month, 2018, including 317,450 trips. Each trip record is mainly composed of the $taxi\_id$, $trip\_start\_timestamp$, $trip\_end\_timestamp$, $trip\_miles$, $pickup/dropoff\_community\_area$, $fare$, etc. In the simulations, we treat the taxi-hailing requests in the trace as the MCS task, and see the drivers as MCS workers. From the trace, we can derive that the requests are distributed in multiple community areas. In order to make it more applicable to our model, we randomly select a community area and focus on dealing with the taxi-hailing requests whose $pickup\_community\_area$ values are the selected area. Here, we select the 8th community area, and the number of taxis and taxi-hailing requests in this community area are 125 and 6349, respectively. Then, the algorithms terminate when the budget runs out or all taxi-hailing requests are handled out. Next, we can derive each driver's travelling distance. Then, we set the value of each driver's cost in proportional to its travelling distance. Finally, since there are no records about the drivers' QoCs, we generate the QoC of each driver as a value randomly sampled from a Gaussian distribution. Each Gaussian distribution is truncated to the interval $[0,1]$. The corresponding mean and standard deviation are randomly sampled from the uniform distribution on $(0,1)$.

In addition, we also use synthetic datasets to test the performances of the implemented algorithms. In order to achieve unbiased performance comparison, we generate 200 different synthetic datasets. In each dataset, we first set the number of workers N as 100. Then, similar to the real-trace data, we sample the QoC of each worker from a Gaussian distribution which is truncated to the interval $[0,1]$. Besides, the workers' costs are randomly sampled from the uniform distribution on $[1,10]$. Then, the algorithms are conducted 1000 times under each synthetic dataset. The outputs are the average results of the algorithms running on the 200 datasets.

Additionally, in order to evaluate the impact of the differential privacy security parameter δ and the budget of recruiting workers B on the performances of the implemented algorithms, we set the values of δ as 0.2, 0.4, 0.6, 0.8 and set B from 1000 to 10000, respectively. Since the performance of DPF also depends on the value of ϵ, we implement the algorithm with $\epsilon = 0.01$, 0.05 and 0.1 for comparison. The detailed parameter settings are shown in Table 3. Furthermore, the algorithms are implemented in Eclipse IDE for Java Developers, and the simulations are performed on a Windows machine with 8GB RAM, Intel(R) Core(TM) i5 2.90GHz CPU.

Performance Metrics: In the simulations, we tracked five performance metrics: the accumulative reward, the average regret, the privacy leakage, and the time efficiency. The average regret is the value of the total regret divided by the budget B. The privacy leakage of an algorithm is used to evaluate how well the privacy of workers' sensitive data is protected by this algorithm. We use the Kullback-Leibler (KL) divergence to measure the privacy leakage, which is defined as follows:

Definition 5.1 (Privacy Leakage). For two input data sequences $\vec{X}_{1:t-1}$ and $\vec{X}'_{1:t-1}$ which differ in at most one time slot, the privacy leakage is computed as

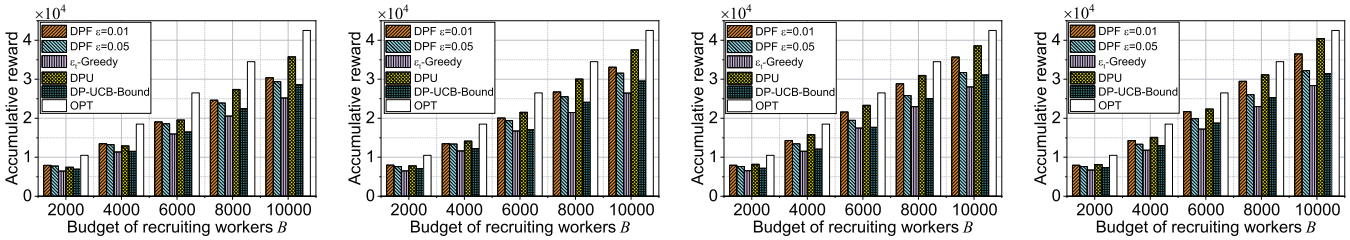
$$\sum_{a_t \in \mathcal{N}} \mathbb{P}\{\Psi_t(\vec{X}_{1:t-1}) = a_t\} \ln\left(\frac{\mathbb{P}\{\Psi_t(\vec{X}_{1:1-t}) = a_t\}}{\mathbb{P}\{\Psi_t(\vec{X}'_{1:1-t}) = a_t\}}\right).$$

Finally, the time efficiency performances refer to the running times of the DPF and DPU algorithms.

5.2 Evaluation Results

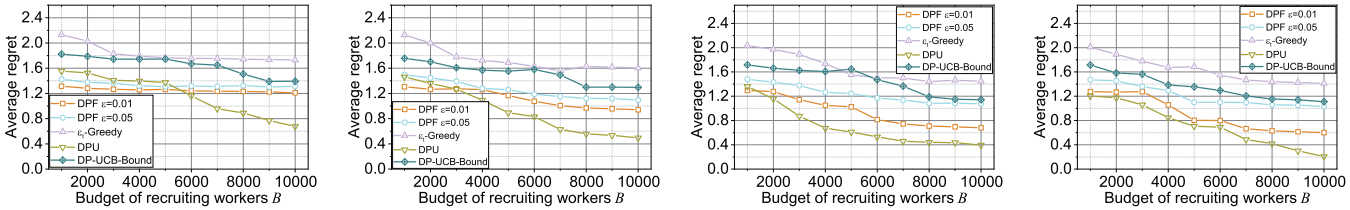
Accumulative Reward: The simulation results of evaluating the accumulative reward performance are plotted in Fig. 4 and Fig. 6. From the results, we can observe that

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.2990221, IEEE Transactions on Mobile Computing



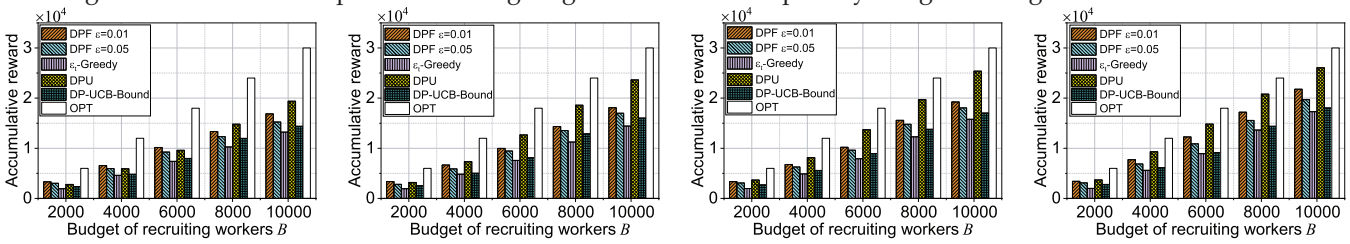
(a) $\delta = 0.2$ (b) $\delta = 0.4$ (c) $\delta = 0.6$ (d) $\delta = 0.8$

Fig. 4: Performance comparisons: accumulative reward vs. differential privacy budget δ using the real-trace dataset



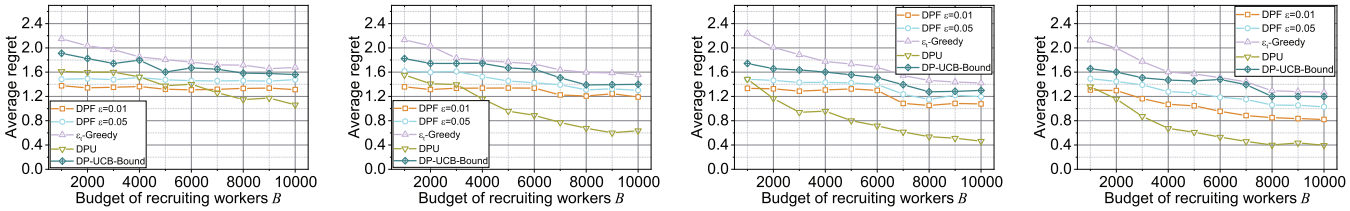
(a) $\delta = 0.2$ (b) $\delta = 0.4$ (c) $\delta = 0.6$ (d) $\delta = 0.8$

Fig. 5: Performance comparisons: average regret vs. differential privacy budget δ using the real-trace dataset



(a) $\delta = 0.2$ (b) $\delta = 0.4$ (c) $\delta = 0.6$ (d) $\delta = 0.8$

Fig. 6: Performance comparisons: accumulative reward vs. differential privacy budget δ using the synthetic datasets



(a) $\delta = 0.2$ (b) $\delta = 0.4$ (c) $\delta = 0.6$ (d) $\delta = 0.8$

Fig. 7: Performance comparisons: average regret vs. differential privacy budget δ using the synthetic datasets

the accumulative rewards of all algorithms increase with the increase of budget B. It is due to the reason that more workers can be recruited with an increasing budget B. Meanwhile, we can also find that a larger δ leads to higher accumulative rewards for all algorithms. This is because that when δ is small, the algorithms have higher privacy levels, which means that the algorithms need to spend more budget identifying the optimal worker. In addition, since DP-UCB-Bound and ϵ_t-Greedy recruit workers without considering their costs, they would recruit workers with large costs, resulting in the rapid consumption of budget and less number of recruited workers. Consequently, DP-UCB-Bound and ϵ_t-Greedy produce less rewards than our proposed algorithms.

Average Regret: The results of evaluating the average regrets based on real-trace dataset and synthetic datasets are shown in Figs. 5 and 7, respectively. From the figures, we can find that when δ increases, the regrets of all implemented algorithms decrease. The results are consistent with the evaluation results of the accumulative rewards. In addition, in most cases, for example, $\delta > 0.2$ and $B \geq 4000$, the average regret performance of DPU is superior to that of DPF, as shown in Figs. 5(b)-5(d) and 7(b)-7(d). However, in the cases when δ and B are small, for example, $\delta \leq 0.2$ and $B < 4000$, a well tuned DPF algorithm outperforms DPU, as shown in Figs. 5(a) and 7(a). More intuitively, when we fix the value of δ, we can discover that as the increasing of the budget B, the DPU algorithm incurs less regret than DPF. These phenomenons are in accordance with the theoretical analyses of regrets in Sections 3.4 and 4.3. Consequently, the MSC platform can choose the suitable algorithm according to practical scenarios.

Privacy Leakage: To evaluate the privacy leakage, we set the differential privacy security parameter δ from 0.1 to 1. In addition, we randomly generated 1000 pairs of data sequences $X_{1:t}$ and $X'_{1:t}$, which differ in at most one record. The results are shown in Fig. 8. We can observe that a larger δ leads to a higher privacy leakage. This is consistent with the definition of differential privacy. Moreover, both of DPF and DPU have low privacy leakage (no larger than 0.15). Here, since the evaluation results based on the real-trace dataset and the synthetic datasets are almost similar, we

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.2990221, IEEE Transactions on Mobile Computing

only present the results produced by using the synthetic datasets.

Time Efficiency: Fig. 9 presents the running time of executing the DPF algorithm and the DPU algorithm under different budget values. We can observe that although the DPU algorithm achieves better performances with regard to the accumulative reward (and the average regret as well) in most cases, the DPU algorithm incurs higher running time compared with DPF. However, the running time of DPU is still less than 2.5s when the budget of recruiting workers B equals to 10000. Therefore, the MSC platform can also take the time efficiency into consideration when choosing the used algorithms.

Remark: Note that for a long-term continuously task, larger budget means more executing time slots. In the above evaluations, we only present the evaluation results of the performances of the accumulative reward and the average regret with the variation of budget. In addition, the results are similar when we increase the executing time slots.

6 Related Works

Currently, it has attracted considerable attention from academia with regard to different research problems in mobile crowdsensing/crowdsourcing systems, including worker recruitment, task allocation, incentive mechanism design, and privacy, etc [1]–[12], [14], [18], [28]–[38]. For example, Z. He et al. in [1] propose a greedy algorithm and a genetic algorithm to solve the worker recruitment problem in vehicle-based crowdsourcing, aiming at maximizing the participation coverage and improving the crowdsourcing quality. L. Yang et al. in [11] develop an auction framework to recruit workers in MCS, and propose a differentially private data aggregation scheme to protect the privacy of workers' sensed data. Nevertheless, most of these existing works conduct the worker recruitment procedure based on the assumption that workers' QoCs are known as a prior. None of them discusses the unknown worker recruitment problem with privacy concern.

So far, MABs have been widely investigated and various MAB policies have also been proposed and utilized into many research fields [19]–[22], [25], [26], [39]–[43]. For instance, [39] has shown that the regret of stochastic MAB grows at least logarithmically over time. In [19], the authors have proposed an index-based policy for stochastic MAB using the UCB policy, and shown that the expected regret of UCB grows at least logarithmically. S. Kang et al. in [43] address the user-channel allocation problem in multi-user multi-channel cognitive radio networks without a prior knowledge of channel statistics, and develop an MAB-based learning algorithm to solve the problem. Compared with our work, the most related works are [21], [26]. In [21], A. C. Y. Tossou et al. propose three UCB-based algorithms for the DP-MAB problem. Nevertheless, this paper has not taken into consideration the costs of pulling arms or budget constraint. Actually, when we introduce the costs and budget constraint into DP-MAB, our DP-MAB problem contains the 0-1 knapsack problems which makes it completely different from that in [21]. Since the knapsack problem is NP-hard, our DP-MAB problem is more challenging. In [26], L. Tran-Thanh et al have propose a budget-limited worker recruitment policy based on MABs. However, they do not take the privacy issue into consideration, which would also increase the difficulties of designing algorithms and analyzing theoretical performances. Overall, none of these existing works combines MAB, differential privacy, and limited budget to solve the unknown worker recruitment problem for MCS systems.

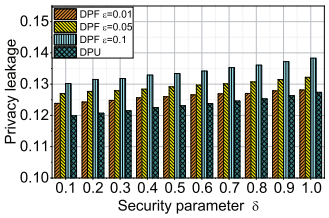


Fig. 8: Privacy leakage vs. δ.

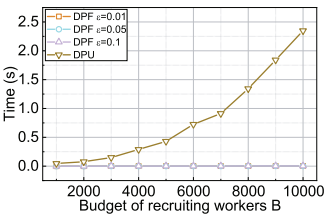


Fig. 9: Running time vs. B.

7 Discussion

In this paper, we mainly focus on handling a single continuous long-term task for the unknown worker recruitment problem. Actually, it can be extended to support the scenario of multiple tasks. Suppose that an MCS platform publicizes some tasks at the same time. First, consider a special case where these tasks and the sets of workers that are willing to perform these tasks are independent with each other. For this case, we can divide the problem into multiple single-task problems and directly apply our proposed algorithms to solve them in parallel. Second, we consider a general case where each of these tasks can be performed by multiple workers and each worker can perform multiple tasks. For this case, we need to recruit a set of workers in each time slot to perform these tasks. This is actually a combinatorial multi-armed bandit problem, in which a set of arms (called an *arm combination*) are pulled simultaneously in each time slot. To address such a problem, we first construct many combinations of workers and treat them as a series of candidate arm combinations. Then, we take each arm combination as a whole and conduct our bandit policies to solve the combinatorial unknown worker recruitment problem. Note that we only need to select one worker combination in each time slot. Thus, it is unnecessary to determine all possible worker combinations. For example, when the UCB policy is adopted, we only need to determine a worker combination with the maximal UCB index value in each time slot. This can be approximately solved by using a greedy selection strategy.

In our DP-MAB model, the workers are assumed to truthfully submit their costs to the platform. As we have not leveraged any auction mechanisms, our algorithms cannot completely guarantee the workers' truthfulness. Even though, this does not mean that the workers can arbitrarily report their costs. Note that our algorithms recruit workers in the descending order of the values of QoC per cost. If a worker increases his/her cost beyond a critical value (which is equivalent to the critical cost in the Second-Price auction), he/she might be replaced by the worker with this critical cost. This implies that the total payment paid by the platform will be no more than that in the case where a truthful Second-Price auction is adopted. Further, if seeking for the complete truthfulness, we must combine our MAB policies with a truthful auction mechanism. However, the

UCB bandit policy cannot work in this case, because it has been proved that the truthful mechanisms must separate "exploration" from "exploitation" [44]. As for the ϵ-First bandit policy, we can directly apply an auction mechanism to our model by adding a payment computation process in each time slot. Moreover, we need to guarantee that the total payment is no more than the budget. It should be pointed out that when involving the hybrid differentially private mechanism, we cannot obtain precise QoC values, and consequently we can only achieve an approximate truthfulness and individual rationality.

8 Conclusion

In this paper, we focus on the differentially private unknown worker recruitment problem in the MCS system. To address this problem, we introduce the Multi-Armed Bandit (MAB) model, and turn the unknown worker recruitment problem into a Differentially Private MAB (DP-MAB) game. Moreover, we propose two budget-feasible differentially private arm-pulling algorithms, i.e., the ϵ-First-based Differentially Private algorithm (DPF) and the UCB-based Differentially Private algorithm (DPU). The proposed DPF and DPU algorithms can not only satisfy δ-differential privacy, but also achieve provable theoretical performance bounds on the expected regrets. Finally, extensive simulations are conducted to verify the significant performances of DPF and DPU.

References

[1] Z. He, J. Cao, and X. Liu, "High quality participant recruitment in vehicle-based crowdsourcing using predictable mobility," in *IEEE INFOCOM*, 2015.

[2] L. Pu, X. Chen, J. Xu, and X. Fu, "Crowdlet: Optimal worker recruitment for self-organized mobile crowdsourcing," in *IEEE INFOCOM*, 2016.

[3] E. Wang, Y. Yang, J. Wu, W. Liu, and X. Wang, "An efficient prediction-based user recruitment for mobile crowdsensing," *IEEE TMC*, vol. 17, no. 1, 2018.

[4] H. Gao, C. H. Liu, J. Tang, D. Yang, P. Hui, and W. Wang, "Online quality-aware incentive mechanism for mobile crowd sensing with extra bonus," *IEEE TMC*, 2018.

[5] C. Qiu, A. C. Squicciarini, S. M. Rajtmajer, and J. Caverlee, "Dynamic contract design for heterogenous workers in crowdsourcing for quality control," in *IEEE ICDCS*, 2017.

[6] D. Zhao, X.-Y. Li, and H. Ma, "Budget-feasible online incentive mechanisms for crowdsourcing tasks truthfully," *IEEE/ACM TON*, vol. 24, no. 2, 2016.

[7] M. Xiao, J. Wu, S. Zhang, and J. Yu, "Secret-sharing-based secure user recruitment protocol for mobile crowdsensing," in *IEEE INFOCOM*, 2017.

[8] Q. Wang, Y. Zhang, X. Lu, Z. Wang, Z. Qin, and K. Ren, "Rescuedp: Real-time spatio-temporal crowdsourced data publishing with differential privacy," in *IEEE INFOCOM*, 2016.

[9] D. Peng, F. Wu, and G. Chen, "Data quality guided incentive mechanism design for crowdsensing," *IEEE TMC*, vol. 17, no. 2, 2018.

[10] J. Li, Y. Zhu, Y. Hua, and J. Yu, "Crowdsourcing sensing to smartphones: A randomized auction approach," *IEEE Transactions on Mobile Computing*, vol. 16, no. 10, pp. 2764–2777, 2017.

[11] L. Yang, M. Zhang, S. He, M. Li, and J. Zhang, "Crowd-empowered privacy-preserving data aggregation for mobile crowdsensing," in *ACM MobiHoc*, 2018.

[12] X. Tang, C. Wang, X. Yuan, and Q. Wang, "Non-interactive privacy-preserving truth discovery in crowd sensing applications," in *IEEE INFOCOM*, 2018.

[13] J. Lin, D. Yang, M. Li, J. Xu, and G. Xue, "Frameworks for privacy-preserving mobile crowdsensing incentive mechanisms," *IEEE TMC*, vol. 17, no. 8, 2018.

[14] J. Lin, M. Li, D. Yang, and G. Xue, "Sybil-proof online incentive mechanisms for crowdsensing," in *IEEE INFOCOM*, 2018.

[15] J. Shu, X. Jia, K. Yang, and H. Wang, "Privacy-preserving task recommendation services for crowdsourcing," *IEEE TSC*, 2018.

[16] R. Zhou, Z. Li, and C. Wu, "A truthful online mechanism for location-aware tasks in mobile crowd sensing," *IEEE TMC*, vol. 17, no. 8, 2018.

[17] Z. Wang, J. Li, J. Hu, Z. Li, and Y. Li, "Towards privacy-preserving incentive for mobile crowdsensing under an untrusted platform," in *IEEE INFOCOM*, 2019.

[18] Z. Wang, J. Hu, R. Lv, J. Wei, Q. Wang, D. Yang, and H. Qi, "Personalized privacy-preserving task allocation for mobile crowdsensing," *IEEE TMC*, vol. 18, no. 6, 2019.

[19] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[20] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Foundations and Trends in Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.

[21] A. C. Y. Tossou and C. Dimitrakakis, "Algorithms for differentially private multi-armed bandits," in *AAAI*, 2016.

[22] N. Mishra and A. Thakurta, "(nearly) optimal differentially private stochastic multi-arm bandits," in *UAI*. AUAI Press, 2015.

[23] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography Conference*, pp. 265–284.

[24] T.-H. H. Chan, E. Shi, and D. Song, "Private and continual release of statistics," *ACM Transactions on Information and System Security*, vol. 14, no. 3, p. 26, 2011.

[25] L. Tran-Thanh, A. Chapman, E. M. de Cote, A. Rogers, and N. R. Jennings, "ϵ-first policies for budget-limited multi-armed bandits," in *AAAI*, 2010.

[26] L. Tran-Thanh, S. Stein, A. Rogers, and N. R. Jennings, "Efficient crowdsourcing of unknown experts using bounded multi-armed bandits," *Artificial Intelligence*, vol. 214, pp. 89–111, 2014.

[27] "Chicago taxi trips," https://www.kaggle.com/chicago/chicago-taxi-trips-bq.

[28] L. Wang, Z. Yu, D. Zhang, B. Guo, and C. Liu, "Heterogeneous multi-task assignment in mobile crowdsensing using spatiotemporal correlation," *IEEE TMC*, vol. 18, no. 1, 2019.

[29] M. Xiao, J. Wu, L. Huang, R. Cheng, and Y. Wang, "Online task assignment for crowdsensing in predictable mobile social networks," *IEEE TMC*, vol. 16, no. 8, pp. 2306–2320, 2017.

[30] L. Xiao, Y. Li, G. Han, H. Dai, and H. V. Poor, "A secure mobile crowdsensing game with deep reinforcement learning," *IEEE TIFS*, vol. 13, no. 1, 2018.

[31] F. Restuccia, P. Ferraro, S. Silvestri, S. K. Das, and G. L. Re, "Incentme: Effective mechanism design to stimulate crowdsensing participants with uncertain mobility," *IEEE TMC*, 2018.

[32] G. Xu, H. Li, S. Liu, M. Wen, and R. Lu, "Efficient and privacy-preserving truth discovery in mobile crowd sensing systems," *IEEE TVT*, vol. 68, no. 4, 2019.

[33] C. Luo, X. Liu, W. Xue, Y. Shen, J. Li, W. Hu, and A. X. Liu, "Predictable privacy-preserving mobile crowd sensing: A tale of two roles," *IEEE/ACM TON*, vol. 27, no. 1, 2019.

[34] C. Miao, W. Jiang, L. Su, Y. Li, S. Guo, Z. Qin, H. Xiao, J. Gao, and K. Ren, "Privacy-preserving truth discovery in crowd sensing systems," *ACM Transactions on Sensor Networks*, vol. 15, no. 1, 2019.

[35] M. Xiao, K. Ma, A. Liu, H. Zhao, Z. Li, K. Zheng, and X. Zhou, "Sra: Secure reverse auction for task assignment in spatial crowdsourcing," *IEEE TKDE*, 2019.

[36] Y. Zhang, Y. Gu, M. Pan, N. H. Tran, Z. Dawy, and Z. Han, "Multi-dimensional incentive mechanism in mobile crowdsourcing with moral hazard," *IEEE TMC*, vol. 17, no. 3, 2018.

[37] Y. Liu, B. Guo, C. Chen, H. Du, Z. Yu, D. Zhang, and H. Ma, "Foodnet: Toward an optimized food delivery network based on spatial crowdsourcing," *IEEE TMC*, vol. 18, no. 6, 2019.

[38] Y. Zhan, Y. Xia, Y. Liu, F. Li, and Y. Wang, "Incentive-aware time-sensitive data collection in mobile opportunistic crowdsensing," *IEEE TVT*, vol. 66, no. 9, 2017.

[39] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in applied mathematics*, vol. 6, no. 1, pp. 4–22, 1985.

[40] S. Buccapatnam, A. Eryilmaz, and N. B. Shroff, "Stochastic bandits with side observations on networks," in *ACM SIGMETRICS*, 2014.

[41] L. Xu, C. Jiang, Y. Qian, Y. Zhao, J. Li, and Y. Ren, "Dynamic privacy pricing: A multi-armed bandit approach with time-variant rewards," *IEEE TIFS*, vol. 12, no. 2, 2017.
[42] D. Zhou and C. Tomlin, "Budget-constrained multi-armed bandits with multiple plays," in *AAAI*, 2018.
[43] S. Kang and C. Joo, "Low-complexity learning for dynamic spectrum access in multi-user multi-channel networks," in *IEEE INFOCOM*, 2018.
[44] M. Babaioff, Y. Sharma, and A. Slivkins, "Characterizing truthful multi-armed bandit mechanisms," in *Proceedings of the 10th ACM Conference on Electronic Commerce*. ACM, 2009, pp. 79–88.

Hui Zhao is a PhD in the School of Computer Science and Technology at the University of Science and Technology of China (USTC) currently. She received her BS degree from the School of Computer Science and Engineering at the Wuhan Institute of Technology, China, in 2016. Her research interests include spatial crowdsourcing, vehicular ad hoc networks, auction theory, and privacy-preserving mechanism.

Mingjun Xiao is a professor in the School of Computer Science and Technology at the University of Science and Technology of China (USTC). He received his Ph.D. degree from USTC in 2004. His research interests include crowdsourcing, mobile social networks, vehicular ad hoc networks, mobile cloud computing, auction theory, data security and privacy. He has published more over 60 papers in referred journals and conferences, including TMC, TC, TPDS, TKDE, TSC, INFOCOM, ICNP, etc. He served as the TPC member of INFOCOM'20, INFOCOM'19, ICDCS'19, DASFAA'19, INFOCOM'18, etc. He is on the reviewer board of several top journals such as TMC, TON, TPDS, TSC, TVT, TCC, etc.

Jie Wu is the Director of the Center for Networked Computing and Laura H. Carnell professor at Temple University. He also serves as the Director of International Affairs at College of Science and Technology. He served as Chair of Department of Computer and Information Sciences from the summer of 2009 to the summer of 2016 and Associate Vice Provost for International Affairs from the fall of 2015 to the summer of 2017. Prior to joining Temple University, he was a program director at the National Science Foundation and was a distinguished professor at Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. Dr. Wu regularly publishes in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Mobile Computing, IEEE Transactions on Service Computing, Journal of Parallel and Distributed Computing, and Journal of Computer Science and Technology. Dr. Wu was general co-chair for IEEE MASS 2006, IEEE IPDPS 2008, IEEE ICDCS 2013, ACM MobiHoc 2014, ICPP 2016, and IEEE CNS 2016, as well as program co-chair for IEEE INFOCOM 2011 and CCF CNCC 2013. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a Fellow of the AAAS and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.

Yun Xu received the PhD degree in computer science from the University of Science and Technology of China (USTC), Hefei, in 2002. He is currently a professor with the School of Computer Science of USTC, a member of National High Performance Computing Center at Hefei, and also the supervisor of PhD students under the collaboration scheme between the City University of Hong Kong and USTC. Now, he is leading a group of research students in some high-performance computing and bioinformatics research. His research interests include bioinformatics, biological sequence analysis and mining, parallel algorithms, and parallel programming model, and performance optimization. He has authored or coauthored more than 50 papers. He is a member of the IEEE and the ACM.

He Huang is a professor in the School of Computer Science and Technology at Soochow University, P.R. China. He received his Ph.D. degree in Department of Computer Science and Technology from University of Science and Technology of China (USTC), in 2011. His current research interests include traffic measurement, spectrum auction, privacy preserving in auction, and algorithmic game theory. He is a Member of both IEEE and ACM.

Sheng Zhang is an associate professor in the Department of Computer Science and Technology, Nanjing University. He is also a member of the State Key Lab. for Novel Software Technology. He received the BS and PhD degrees from Nanjing University in 2008 and 2014, respectively. His research interests include cloud computing and edge computing. To date, he has published more than 60 papers, including those appeared in *TMC*, *TPDS*, *TC*, *MobiHoc*, *ICDCS*, *INFOCOM*, *IWQoS*, and *ICPP*. He received the Best Paper Runner-Up Award from IEEE MASS 2012. He is a member of the IEEE and a senior member of the CCF.