Multi-Armed Bandits Based Task Selection of A Mobile Crowdsensing Worker

Qinghua Sima[†], Guoju Gao[†*], He Huang[†*], Yu-E Sun[‡], Yang Du[†], Xiaoyu Wang[†], and Jie Wu[§]
[†]School of Computer Science and Technology, Soochow University, China
[‡]School of Rail Transportation, Soochow University, China
[§]Department of Computer and Information Sciences, Temple University, USA
[*]Correspondence to: gjgao@suda.edu.cn, huangh@suda.edu.cn

***Abstract*—As the popularity of mobile devices continues to increase, Mobile Crowdsensing (MC), a scalable and efficient data collection method, has received widespread attention. Although lots of effort has been devoted to studying the task assignment or worker recruitment in MC, most of them focus on how to maximize the profit from the perspective of the platform while ignoring rational individual workers' entitlement. We creatively start from the worker's perspective to find the task selection strategy to maximize the worker's profit. In this paper, the problem of unknown task selection is modeled as a Multi-Armed Bandit (MAB), on which three types of additional constraints are considered. The first constraint is the device budget. Workers choose and conduct tasks before it is exhausted. The second constraint is the personal preference regarding the traveling cost. The third constraint is the balance requirement of the MC platform, which has regulations on the tasks' execution rounds. In addition to the dilemma between exploration and exploitation in the classical MAB, we have to face the tradeoff between the reward and all the constraints above. To this end, we first adopt the epoch-style algorithm to reduce the number of switches between any two sensing tasks and further build new algorithms to deal with different constraints. The traveling cost and platform balance are involved in the task index computation as a penalty. We conduct extensive simulations based on real-world traces to verify the significant performance of our proposed algorithms.**

***Index Terms*—multi-armed bandits, mobile crowdsensing, task selection, device budget, preference, balance**

I. Introduction

Mobile Crowdsensing (MC) [1]–[5] has become a competitive public data collection method with the development of mobile smart devices and their embedded sensors, which often can sense, calculate, and transmit data. The MC platform recruits specific workers to conduct sensing tasks, through which it can collect data and finally realize the purpose of data analysis or application, such as Waze [6], Amazon MechanicalTurk [7], *etc*.

In the MC system, three directions are the most critical: task assignment [8] or worker recruitment [9], incentives design [10], and privacy protection [11]. However, most of the existing studies neglect workers' entitlement. In such a case, the individual worker might be unwilling to participate as a selfish and rational person in the MC process. In this paper, we focus on the task selection problem from the perspective of an individual worker, and the MC process is shown in Fig. 1. The MC platform publishes a set of location-based sensing tasks within a specific geographic area and informs

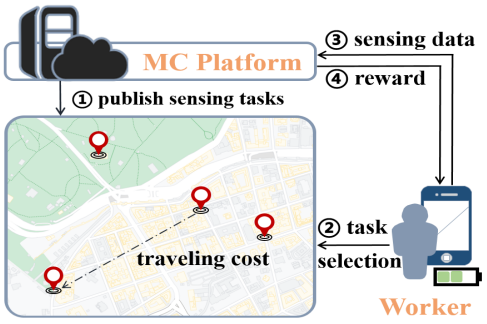


Fig. 1. Illustration of MC process from the perspective of a worker.

workers to select and execute tasks. The worker carrying the smart device will go to the location of a particular task and collect the required data, return it to the platform, and then get the corresponding reward. Note that the traveling cost will be generated because of workers' movements when switching tasks. In the MC system, time is slotted, and we use the term "round" to denote the time slot. At the beginning of each round, the worker should select one task to execute and then go to the location to conduct it. In this paper, we concentrate on how to design efficient task selection strategies to assist these workers in demand.

Most existing researches [8], [12] assume that the reward information for completing tasks is known in advance and believe that there is no traveling cost when switching tasks. We consider a more practical situation in which the reward information of all the tasks follows an unknown distribution. Additional, each time the worker switches from one task to another, the movement will certainly produce the traveling cost. In this case, maximizing the worker's total profit (*i.e.*, the difference between the total achieved reward and the total traveling cost) has become our focus.

Since the workers are unaware of the corresponding reward information for the tasks, we adopt the reinforcement learning technique [13] to learn it. We further consider the possible constraints: 1) each device is always equipped with quantitative resources, which means the total resources consumed by the device when conducting tasks cannot exceed its budget capacity; 2) when the worker switches from one task position to another, the traveling cost is produced due to position transfer; 3) the MC platform also requires workers to meet

a certain balance in execution rounds. The term "balance" means that the platform wants to avoid the situation in which some sensing tasks are over-executed while the others are under-executed. In practical scenarios, the above constraints must be considered because resource-exhausted devices can't help with task execution, and a worker's task selection is inevitably affected by the platform's requirements and his distance to the task. We model the problem as a constrained Multi-Armed Bandit (MAB). The worker must deal with the dilemma between exploitation (*i.e.*, selecting the task with the highest empirical reward so far) and exploration (*i.e.*, trying other tasks to discover the potentially optimal task), and face the trade-off between the reward and the three constraints. Note that our optimization goal is to maximize the total profit of a worker, which is somewhat different from the goal of the traditional MAB problem (maximize the total reward).

Only a few studies [14]–[16] have combined the MAB model with the MC process in which the reward information for task completion is unknown. For instance, [14] focuses on worker recruitment under a limited budget to maximize task completion quality. Reference [15] proposes a MAB-based worker recruitment framework to address the issues of lacking prior knowledge about workers' quality and tasks' ground truth in spatial crowdsourcing. All these researches have the problem of ignoring workers' rights and failing to consider the actual constraints fully. To bridge this gap, we focus on protecting individual workers' rights and dealing with actual constraints effectively. That is, we consider the task selection problem under various constraints from a worker's perspective.

Intuitively, to maximize the worker's profit, the worker should maximize the total reward while minimizing the total traveling cost. The epoch-style algorithm (such as the UCB2 algorithm [13]) where all rounds are divided into many epochs with increased sizes is suitable to solve our problem. At the beginning of each epoch, the worker will select a sensing task according to a pre-established standard, and then conduct it in the whole epoch (multiple rounds). In this way, the worker can reduce the number of task switches to decrease the total traveling cost. Now, the focus is on how to establish a selection standard. We first adopt the Upper Confidence Bound (UCB) to denote the UCB-based reward for all sensing tasks; then, we use the ratio of the UCB-based reward and the average resource consumption as the selection standard. Next, we consider the traveling-cost-related preference of the worker and balance requirement of the platform in the index computation. We design a preference-aware task selection algorithm and a balance-aware task selection algorithm, respectively.

We highlight the main contributions as follows:

- We break away from the inherent research framework from the perspective of the MC platform and design a task selection mechanism that seeks to maximize a worker's total profit. This perspective supplements the missing angles in the previous state-of-the-art work on mobile crowdsensing.
- We adopt the epoch-style strategy for our problem. We first apply the ratio of the UCB-based reward and the average resource consumption in the computation of the index for each task, and then take the traveling-cost-related preference as a penalty into the index computation and devise another algorithm.
- We view the MC process not just in isolation from a worker's perspective, but also consider the platform's requirements. That is, workers should meet the platform's expectation for the number of rounds of task execution (called balance). We adopt virtual queue technology and propose an extended algorithm on the basis.
- We conduct extensive simulations based on real-world traces to evaluate the significant performance of the proposed algorithms under various constraints.

The remainder of the paper is organized as follows. We first present the system model and the optimization problem in Section II. Next, we adopt the epoch-style algorithm for the task selection problem in Section III. We further devise a preference-aware algorithm in Section IV and a balance-aware algorithm in Section V. In Section VI, we evaluate the performance of our algorithms. After reviewing the related work in Section VII, we conclude the paper and discuss future research directions in Section VIII.

II. Model & Problem Formulation

A. System Model

We consider a mobile crowdsensing system consisting of a platform and many workers. Our goal is to design available strategies for an individual worker to select tasks. Suppose that every worker in the MC system is rational and selfish. The worker always wants to take action solely to maximize his profit under a certain device budget. Since the reward information is unknown, each worker must learn it during the process of conducting tasks. In such settings, the worker might fall into the dilemma between exploitation and exploration as studied in the MAB model. Note that our model differs from the traditional MAB model due to the constraints of the device budget and the traveling cost. Here, the device budget is a value that reflects the total number of all the limited resources that a smart device can use. It should be considered because some resources will be consumed when the worker conducts the sensing tasks, such as battery energy. The traveling cost indicates the price when switching from one task to another and must be considered because sensing tasks are always bound to specific locations in the MC system. We reasonably believe that the traveling cost depends on the distance between two sensing tasks. The goal of the worker is to maximize the total profit under specific constraints. In the MC system, we suppose that the resource consumption of the device is negligible when switching tasks.

Let t denote the time slot (called "round"). In each round, a worker can choose only one task to conduct, and the execution time for each task is equal. Based on this, we stipulate the location and reward information about sensing tasks. Consider that there are total m heterogeneous location-based sensing tasks distributed in the urban area. Let $\mathcal{S} = \{s_1, \cdots, s_i, \cdots, s_m\}$ denote the set of sensing tasks. The term "heterogeneous"

means that the reward and resource consumption of conducting these tasks are different. The term "location-based" shows that a worker must go to a specific location to conduct one task. For each task s_i, we use $l_i \in \mathcal{L}$ to denote its location, and all discrete locations form $\mathcal{L}$. We let l_i^t denote the location of the worker in round t when he conducts the task s_i in l_i.

In this paper, the *unknown* tasks are taken into consideration in the MC system; "unknown" means that the worker has no knowledge about the reward information of tasks. We use a normalized nonnegative random variable $r_i^t \in (0, 1]$ to denote the reward of the worker completing the task s_i in the t-th round. For $\forall s_i \in \mathcal{S}$, $\{r_i^1, r_i^2, \cdots, r_i^t, \cdots\}$ follows an unknown independent and identically distribution with an unknown expectation r_i. Moreover, if the worker does not select the task $s_i \in \mathcal{S}$ in round t, we have $r_i^t = 0$; else, after the worker completes the task i, the reward r_i^t will be revealed. At the same time, when the worker conducts the sensing task s_i in round t, the resource consumption is determined and is denoted as b_i^t. For $\forall s_i \in \mathcal{S}$, $\{b_i^1, b_i^2, \cdots, b_i^t, \cdots\}$ follows an unknown independent and identically distribution with an unknown expectation b_i. Here, the value of b_i^t is normalized into the range $(0, 1]$. We use the notation $\mathcal{B}$ to denote the limited device budget value. The stop condition of a worker's task selection process is that the tasks no longer need to be conducted or the required device resources exceed the remaining device budget. Due to the limited device budget $\mathcal{B}$, the task selection round for the worker is finite. Assume $t(\mathcal{B})$ denotes the finite time. We consider that the device budget $\mathcal{B}$ is sufficient to support a worker to complete all the tasks at least once.

Our next focus is the notation of the traveling cost between any two tasks in the MC system. When a worker completes his current task s_i, he will have two choices: 1) still selecting the same task; 2) switching to another task (*e.g.*, $s_j \in \mathcal{S}$). In the first case (*i.e.*, $l_i^{t-1} \rightarrow l_j^t$, where $i = j$) , the worker does not need to travel, so no traveling cost is generated. While in the second case (*i.e.*, $l_i^{t-1} \rightarrow l_j^t$, where $i \neq j$), the worker must travel from l_i to another location l_j to conduct the task s_j. This traveling process will inevitably result in some extra cost, called *traveling cost*. Let c_{ij} denote the traveling cost from l_i to l_j, positively correlated with the distance between tasks. We consider that traveling costs are symmetric, that is, $c_{ij} = c_{ji}$ and $c_{ii} = 0$. For simplicity, the values of traveling cost are mapped into the interval $[0, 1]$.

B. Problem Formulation

In the MC system with location-based unknown sensing tasks, a worker's goal is to maximize his total profit under the device budget $\mathcal{B}$ after considering the traveling cost. We can regard the above problem as an online learning problem, model it with a multi-armed bandit under device budget constraint, and combine the above notations to formulate the optimization goal as follows:

$$Maximize: \mathbb{E}\Big[\sum_{t=1}^{t(\mathcal{B})}(\sum_{s_i \in \mathcal{S}} r_i^t \cdot \mathbb{I}_i^t - \sum_{s_i, s_j \in \mathcal{S}} c_{ij} \cdot \mathbb{I}_i^{t-1} \cdot \mathbb{I}_j^t)\Big] \quad (1)$$

TABLE I
DESCRIPTION OF MAJOR NOTATIONS.

Variable	Description
m	the number of sensing tasks.
t	the index for rounds (time slots).
s_i, $\mathcal{S}$	the indexes and the set of all sensing tasks.
l_i, $\mathcal{L}$	the location of s_i and the set of all locations.
$\mathcal{B}$	the device budget.
$\mathcal{B}_t$	the remaining device budget till round t.
b_i	the resource consumption distribution's mean of s_i.
b_i^t	the resource consumption of s_i in round t.
$\overline{b}_i(t)$	the average resource consumption of s_i till round t.
$t(\mathcal{B})$	the total rounds under the device budget $\mathcal{B}$.
r_i	the reward distribution's mean of s_i.
r_i^t	the reward of conducting s_i in round t.
$\overline{r}_i(t)$	the average reward of s_i till round t.
c_{ij}	the cost of the worker traveling from l_i to l_j.
$R_i(t)$	total rounds of s_i conducted till round t.
$E_i(t)$	total epochs of s_i conducted till round t.
$p_{ij}(t)$	the penalty of traveling-cost-related preference between s_i and s_j in round t.
e_i	the expected minimum execution fraction of s_i.
ϱ_1, ϱ_2	the balance parameters in (14) and (17).
$Q_i(t)$	the virtual queue length of Q_i in round t.

$$Subject\ to: c_{ij} = 0 \ \text{ for } \ l_i^{t-1} \rightarrow l_j^t \text{ where } i = j \quad (2)$$

$$\sum_{t=1}^{t(\mathcal{B})} \sum_{s_i \in \mathcal{S}} \mathbb{I}_i^t \cdot b_i^t \leq \mathcal{B} \quad (3)$$

$$\sum_{s_i \in \mathcal{S}} \mathbb{I}_i^t = 1 \ \text{ for } \ \forall t \geq 1 \quad (4)$$

$$\mathbb{I}_i^t \in \{0, 1\} \ \text{ for } \ \forall t \geq 1, s_i \in \mathcal{S} \quad (5)$$

Assume $\mathbb{I}_i^t$ for $\forall t \geq 1, s_i \in \mathcal{S}$ denotes the decision variable, where $\mathbb{I}_i^t = 1$ means that the worker will conduct the task s_i in round t; otherwise, $\mathbb{I}_i^t = 0$. This reflects (5), which means that at the same time, any task only has two states to be executed and not executed.

Besides, (2) means that no traveling cost is generated when the worker does not change his selected sensing task in two consecutive rounds. Equation (3) denotes that the total consumed resource cannot exceed the limited device budget, and (4) indicates that in each round, only one task is conducted by the worker.

III. DESIGN OF EPOCH-BASED ALGORITHM

A. The Constraint on Device Budget

In the MC system, workers are equipped with specific smart devices to complete sensing tasks. Therefore, the possible constraints of smart devices cannot be ignored. Since the device consumes some non-reusable resources in conducting tasks, a device budget $\mathcal{B}$ always exists. In task selection and execution, the device budget is continuously consumed. After $\mathcal{B}$ is exhausted, the worker will no longer be able to select and execute tasks. The device budget has the following two fundamental properties.

First of all, when workers conduct tasks, the resource consumption of the device is random. Battery energy is an excellent example to illustrate this property. Many different

Algorithm 1 Epoch-Based Solution (EBS) for Task Selection

Require: α, $\mathcal{S}$, $\mathcal{B}$, and c_{ij} for $\forall s_i, s_j \in \mathcal{S}$

for $t = 1 : m$ **do**
  The worker conducts the task $i_t = s_t$, and gets r_i^t.
  Update: $\overline{r}_i(t)$, $\overline{b}_i(t)$, $R_i(t)$, $E_i(t)$.
end for
Let $t \Leftarrow m+1$, $\mathcal{B}_t = \mathcal{B} - \sum_{k=1}^{m} b_k^k$.
while $true$ **do**
  Select the task with the highest ratio in (10),
  $i_{now} = \underset{s_i \in \mathcal{S}}{\operatorname{argmax}} \left((\overline{r}_i(t) + \sqrt{\frac{(1+\alpha)\ln(\frac{et}{\lceil \tau(E_i(t)) \rceil})}{2\lceil \tau(E_i(t)) \rceil}}) / \overline{b}_i(t) \right)$.
  Calculate the epoch length for the selected task i_{now}, $l(\tau) = \lceil \tau(E_{i_{now}}(t)+1) - \tau(E_{i_{now}}(t)) \rceil$.
  Let $T \Leftarrow t + l(\tau)$.
  while $t \leq T$ **do**
    $\mathcal{B}_t \Leftarrow \mathcal{B}_t - b_{i_{now}}^t$.
    if $\mathcal{B}_t \leq 0$ **then**
      Terminate.
    end if
    Go to the location of i_{now}, conduct it for one round.
    $t \Leftarrow t + 1$.
    Update: $\overline{r}_i(t)$, $\overline{b}_i(t)$, $R_i(t)$, $E_i(t)$, $\widehat{r}_i(t)$.
  end while
end while

factors work together to cause random energy consumption of the battery. For example, the factors include not only the type and number of sensors required to conduct sensing tasks but also the local network conditions. The unpredictability of these factors brings difficulties to qualitatively and quantitatively regulating battery energy consumption. The randomness of device resource consumption can be intuitively derived from the randomness of battery energy. Second, we believe that the resource consumption of the device only takes place in the process of conducting tasks. In the process of position transfer, the resource consumption of the device can be ignored.

For conducting any task s_i, the resources consumed by the device in different rounds follow a specific distribution. This distribution is unknown to the workers and needs to be learned through the task execution process. Let b_i denote the expected mean of the resource consumption distribution for a particular task s_i. Let b_i^t denote the resource consumption of executing the task s_i in round t, and this value lies in the interval $(0, 1]$.

Assuming that we only focus on maximizing the worker's reward, it is very likely that we will end the process of executing tasks early due to insufficient resources. Assuming that we only blindly conduct tasks that consume the least amount of resources, our cumulative reward may be small, contrary to our prescribed workers' rational characteristics. Therefore, the basic solution will be our focus.

B. Basic Solution

Unlike the traditional MAB problem, which only deals with the dilemma between exploration and exploitation, we must also face the trade-off between the reward and traveling cost. It is worth noting that although some tasks are well rewarded, they may not be the best choice because the traveling cost to move there from the worker's current location may be high.

To this end, we intend to design a selection strategy to maximize the total profit of the worker by maximizing the total reward and minimizing the total traveling cost. The UCB strategy has good reward performance in a dilemma between exploration and exploitation. At the same time, to minimize the total traveling cost, we should try to reduce the number of times the worker switches sensing tasks. The epoch-based strategy (*e.g.*, the UCB2 algorithm) is suitable here. However, it does not involve the device budget constraint during the task selection process. Therefore, we adopt the idea of epoch introduced by the UCB2 algorithm and took the resources consumed by the device into consideration when devising a new index computation for selecting tasks.

We first introduce the exponential function to help to determine the length of an epoch, as follows:

$$\tau(x) = (1+\alpha)^x, \tag{6}$$

where $0 < \alpha < 1$ is a parameter.

Then, we let $R_i(t)$ and $E_i(t)$ for $\forall s_i \in \mathcal{S}$ denote the numbers of rounds and epochs with which the sensing task s_i is conducted up to round t, respectively. According to this, the length of the epoch during which the task s_i would be conducted is defined as follows:

$$l(\tau) \leq \lceil \tau(E_i(t)+1) - \tau(E_i(t)) \rceil. \tag{7}$$

The impact of the device budget constraint on task execution cannot be ignored when calculating the epoch's length. Suppose the resources of the device carried by the worker are exhausted before the current epoch ends. In that case, the current epoch should be terminated immediately, and the inequality relationship in (7) holds.

How to determine the sensing task at the beginning of each epoch becomes our next focus. Assume $\overline{r}_i(t)$ denotes the average reward of conducting the task s_i up to the round t, which is calculated as follows:

$$\overline{r}_i(t) = \left(\sum_{k=1}^{t} r_i^k \cdot \mathbb{I}_i^k \right) / R_i(t). \tag{8}$$

Let $\widehat{r}_i(t)$ including the average reward $\overline{r}_i(t)$ and an adjustment item denote the UCB-based index for each sensing task s_i up to the round t, *i.e.*,

$$\widehat{r}_i(t) = \overline{r}_i(t) + \sqrt{\frac{(1+\alpha)\ln(\frac{et}{\lceil \tau(E_i(t)) \rceil})}{2\lceil \tau(E_i(t)) \rceil}}, \tag{9}$$

where e is the mathematical constant and the parameter α is less than 1. The adjustment item is used to increase the selection probability of rarely chosen tasks.

The selection strategy is to always select the task with the highest ratio of the UCB-based reward and the average resource consumption at the beginning of each epoch. For simplicity, we let i_{now} denote the index of the selected sensing task in the first round t of the current epoch. That is, we have:

$$i_{now} = \operatorname{argmax}_{s_i \in \mathcal{S}} \ \widehat{r}_i(t) / \overline{b}_i(t), \tag{10}$$

where $\overline{b}_i(t)$ means the average resource consumption of conducting the task s_i until round t.

C. Detailed Algorithm

According to the above solution, we propose the epoch-based algorithm (*i.e.*, Epoch-Based Solution for Task Selection, EBS) for the worker, as shown in Alg. 1.

In Steps 1-4, the worker will conduct all tasks once to initialize the value of $\overline{r}_i(t)$. When the total resource consumption does not exceed the limited device budget $\mathcal{B}$, the worker will select one task to conduct in Steps 6-19. More specifically, at the beginning of each epoch, the worker selects the sensing task according to the ratio of the UCB-based reward and the average resource consumption, then computes the epoch's length in Steps 7-8. After task selection, the worker must compare the resources they continuously need with the remaining budget in each round of the current epoch. The task selection algorithm will terminate if the remaining budget cannot cover the resource consumption, as shown in Steps 11-14. Otherwise, the worker would go to the specific location of the selected task and conduct it in Step 15 until the end of the current epoch. The corresponding information and the remaining device budget are updated in Steps 11, 16-17.

EBS combines the UCB-based reward and the heterogeneous resource consumption, so it is quite challenging to analyze the specific regret bound. When the resource consumption for all sensing tasks is uniform, *e.g.*, $b_i^t = 1$ for $\forall s_i \in \mathcal{S}$ and $\forall t \geq 1$, the task index computation in EBS is the same as that in the UCB2 algorithm. In such case, the regret performance is analyzed in [13], which is stated as follows:

$$\sum\nolimits_{i:\Delta_i>0} \Big(\frac{\varphi_1 \ln(2e\mathcal{B}\Delta_i^2)}{2\Delta_i} + \frac{\varphi_2}{\Delta_i} \Big), \tag{11}$$

where $\begin{cases} \varphi_1 = (1+\alpha)(1+4\alpha), \\ \varphi_2 = 1 + \frac{(1+\alpha)e}{\alpha^2} + \Big(\frac{1+\alpha}{\alpha}\Big)^{1+\alpha} \Big(1 + \frac{11(1+\alpha)}{5\alpha^2 \ln(1+\alpha)}\Big). \end{cases}$

and Δ_i means the reward difference between the optimal task s_{i^*} which has the highest mean reward and the i-th sub-optimal task s_i, *i.e.*, $\Delta_i = r_{i^*} - r_i$. Note that the bound for reward regret holds for the certain condition, that is, $\mathcal{B} \geq \max_{i:\Delta_i>0} \frac{1}{2\Delta_i^2}$. Because our objective is to maximize the worker's profit, the regret in our problem consists of two parts: the reward loss (due to unknown reward distribution) and the traveling cost loss. Here, (11) shows the reward loss. The traveling cost loss depends on the number of switches among all sensing tasks. In future work, we will try to give the specific regret bound of EBS.

IV. Preference-Aware Algorithm

A. The Constraint on Preference

In the paper, we assume that workers are rational and selfish, which means that their decision-making process serves to maximize the personal profit. However, in actual situations, the connotation of personal preferences will be more complicated. For this reason, we reasonably consider that the worker has two types of preferences except for rationality: 1) no matter where the worker is currently located, he hopes that the location of the next task to be conducted is closer to the current location, and 2) no matter what tasks the worker has conducted before, he hopes that the next task to be conducted corresponds to as many execution rounds as possible so that the worker can avoid frequent movements.

Algorithm 2 Preference-Aware Solution (PAS)

Require: α, $\mathcal{S}$, $\mathcal{B}$, ϱ_1, and c_{ij} for $\forall s_i, s_j \in \mathcal{S}$

1: **Initialization:** Go to the location of each task and conduct it for one round. Update related information.
2: **while** $true$ **do**
3: Select the task with the highest value in (14), $i_{now} = \text{argmax}_{s_i \in \mathcal{S}} \Big(\widehat{r}_i(t)/\overline{b}_i(t) - \varrho_1 p_{i_{old}i}(t) \Big)$.
4: Calculate the epoch length for the selected task i_{now}, $l(\tau) = \lceil \tau(E_{i_{now}}(t)+1) - \tau(E_{i_{now}}(t)) \rceil$.
5: Go to the location of i_{now}, conduct it and update related information until the $l(\tau)$-th round when $\mathcal{B}_t \geq 0$, otherwise terminate.
6: **end while**

Combining the above analysis, we make the following two assumptions. First, for one task, the higher the traveling cost required to transfer from the current task to it, the less willing the worker is to conduct it. Second, the more the execution rounds of the selected task, the more the worker intends to conduct it to avoid switching positions for a longer time.

Therefore, when computing the index for each sensing task at the beginning of each epoch, the worker should subtract the traveling-cost-related preference penalty from the UCB-based index. Note that the penalty related to one task is highly associated with the traveling cost from the worker's current location and correlated to the number of possible execution rounds. Here, we let β_i denote the number of execution rounds of task s_i when it was selected last time. According to the definition of the epoch, the number of corresponding rounds should have a non-decreasing nature. Therefore, β_i can reflect the execution round of s_i when chosen next time. Based on this, we define the penalty regarding the traveling-cost-related preference constraint, denoted as $p_{ij}(t)$, as follows:

$$p_{ij}(t) = \frac{c_{ij}}{\beta_j}, \quad \text{for } \forall s_i, s_j \in \mathcal{S}. \tag{12}$$

Excessive task switching will lead to the accumulation of total traveling cost and the decrease of total profit, whereas always conducting tasks within a small geographic area will damage the total reward for the worker. Therefore, we introduce a basic solution to make a trade-off between the reward and the traveling cost.

B. Basic Solution

Because of the traveling-cost-related preference constraint, the following adjustment should be made to the index calculation when selecting tasks based on the above definition. Let s_i denote the current sensing task that the worker conducts in round $t-1$, *i.e.*, $i_{t-1} = s_i$. We define a new index for all tasks in round t, denoted as $\widetilde{r}_j(t)$, that is:

$$\widetilde{r}_j(t) = \begin{cases} \widehat{r}_j(t)/\overline{b}_j(t); & j = i, \\ \widehat{r}_j(t)/\overline{b}_j(t) - \varrho_1 \cdot p_{ij}(t); & j \neq i, \end{cases} \tag{13}$$

where $\widehat{r}_j(t) = \overline{r}_j(t) + \sqrt{\frac{(1+\alpha)\ln(\frac{et}{\lceil \tau(E_j(t)) \rceil})}{2\lceil \tau(E_j(t)) \rceil}}$ means the UCB-based reward and the parameter $\varrho_1 > 0$ is used to balance $\widehat{r}_j(t)/\overline{b}_j(t)$ and penalty related to the preference.

In this way, the worker will have more incentives to stay in the same place so that the number of switches among sensing tasks will be decreased. For simplicity, we let i_{old} denote the selected sensing task in the last epoch. Based on this, we focus on how to select the task in the current epoch (*i.e.*, i_{now}). i_{now} is determined as follows:

$$i_{now} = \text{argmax}_{s_i \in \mathcal{S}} \left(\widehat{r}_i(t)/\overline{b}_i(t) - \varrho_1 p_{i_{old}i}(t) \right). \quad (14)$$

According to the above solution, we propose the preference-aware algorithm (*i.e.*, Preference-Aware Solution for Task Selection, PAS) for the workers, as shown in Alg. 2. In Step 1, the worker goes to each task's location and conducts it for one round, through which the task-related information is initialized. Then, the execution of tasks is no longer in the round unit but in the epoch unit. In the first round of each epoch, the worker selects a task according to (14) and calculates the appropriate epoch length, as shown in Steps 3-4. Then, in Step 5 the worker goes to the location of the task i_{now} and executes it continuously. When the current epoch ends, the worker will re-select a task to conduct and start a new epoch. If the device budget is exhausted, the algorithm will terminate regardless of whether the current epoch is over.

C. A Walk-through Example

We present a walk-through example to show the task selection procedure to help understand PAS.

Suppose that there are 3 sensing tasks in the MC system and the device budget is $\mathcal{B} = 10$. We consider that the rewards of each task follow the uniform distribution and in $(0, 1]$, and the expected rewards are $r_1 = 0.3$, $r_2 = 0.5$, $r_3 = 0.7$. Assume that the traveling costs between tasks follow $c_{12} = c_{21} = 0.6$, $c_{13} = c_{31} = 0.3$, and $c_{23} = c_{32} = 0.9$. For simplicity, we let the resource consumption per round for all tasks be the same, *i.e.*, $b_i^t = 1$ for $\forall s_i \in \mathcal{S}$ and $\forall t \geq 1$. Finally, let $\alpha = 0.5$, $\varrho_1 = 1$.

The rewards of tasks in each round are shown in Fig. 2. We assume that the initial location of the worker is l_1. In the initial phases of PAS, the worker will select each task once in the first three rounds. The revealed rewards in each round are marked in blue. Here, the average empirical rewards are initialized. Note that now we let $E_i(t) = 1$ for $1 \leq i \leq 3$.

In the next phases, the rounds are divided into epochs. At the beginning of the first epoch, we have $t = 4$ now, and the worker's current location is l_3. Based on this, the following values can be calculated according to (14).

$$\widehat{r}_1(t)/1 - p_{31}(t) = 0.2 + \sqrt{\frac{1.5\ln(4e/2)}{2*2}} - \frac{0.3}{1} = 0.697;$$
$$\widehat{r}_2(t)/1 - p_{32}(t) = 0.5 + \sqrt{\frac{1.5\ln(4e/2)}{2*2}} - \frac{0.9}{1} = 0.397;$$
$$\widehat{r}_3(t)/1 - p_{33}(t) = 0.8 + \sqrt{\frac{1.5\ln(4e/2)}{2*2}} - \frac{0.0}{1} = 1.597.$$

	initialization			epoch	epoch		...
r_i^t	**t = 1**	**t = 2**	**t = 3**	**t = 4**	**t = 5**	**t = 6**	...
$i_{now} = s_1$	0.2	0.3	0.4	0.7	0.1	0.3	...
$i_{now} = s_2$	0.4	0.5	0.6	0.2	0.8	0.6	...
$i_{now} = s_3$	0.6	0.7	0.8	0.7	0.2	0.5	...

Fig. 2. A walk-through example for the task selection procedure.

Thus, in the current epoch, the worker would select the task s_3. Next, the worker will compute the length of the epoch, that is, $l(\tau) = \lceil 1.5^2 - 1.5^1 \rceil = 1$. After the worker conducts the task s_3, its reward is revealed in Fig. 2, *i.e.*, $r_3^4 = 0.7$. We will update the information, such as $t = 5$, $\mathcal{B}_t = 6$, $E_3(t) = 2$, and $\overline{r}_3(t) = 0.75$. Then, the process continues:

$$\widehat{r}_1(t)/1 - p_{31}(t) = 0.2 + \sqrt{\frac{1.5\ln(5e/2)}{2*2}} - \frac{0.3}{1} = 0.748;$$
$$\widehat{r}_2(t)/1 - p_{32}(t) = 0.5 + \sqrt{\frac{1.5\ln(5e/2)}{2*2}} - \frac{0.9}{1} = 0.448;$$
$$\widehat{r}_3(t)/1 - p_{33}(t) = 0.75 + \sqrt{\frac{1.5\ln(5e/3)}{2*3}} - \frac{0.0}{1} = 1.365.$$

Since $\widehat{r}_3(t)/1 - p_{33}(t)$ has the largest value at this moment, the worker will still select the task s_3 in the next epoch. The length of the epoch is $l(\tau) = \lceil 1.5^3 - 1.5^2 \rceil = 2$. Thus, the worker will conduct the task s_3 in the next 2 rounds, as shown in Fig. 2. The following procedures are omitted and the algorithm terminates when $\mathcal{B}_t \leq 0$.

V. Balance-aware Algorithm

A. The Constraint on Balance

The MC system hopes to avoid the scenario where some tasks are over-executed, but others may be under-executed. If some tasks are over-executed, the platform will receive redundant data. For the platform, this type of data has two main types of defects. First, they cannot increase the validity of data and may even bias the information and conclusions finally obtained by the platform. Second, when workers pass back redundant data, the platform needs to pay for them. The platform also needs to pay more data storage and processing costs. If some tasks are not conducted adequately, the relevant information involved in the collected sensing data is not comprehensive, leading to unreasonable decision-making. For the publisher of tasks, although the fewer data collected, the less reward he needs to pay, these reduced payments cannot make up for the lack of information collected and the loss of erroneous conclusions.

In summary, the MC platform has certain data requirements, and we need to convert them into a specific constraint. We introduce a parameter e_i to denote the required minimum fraction of rounds in which the worker would conduct the task s_i. We can formalize the balance constraint as follows:

$$\liminf_{B \to \infty} \mathbb{E}[R_i(t(\mathcal{B}))] \geq t(\mathcal{B}) \cdot e_i, \text{ for } \forall s_i \in \mathcal{S}, \quad (15)$$

where $t(\mathcal{B})$ means the total executable rounds under the device budget $\mathcal{B}$ and $R_i(t)$ is the total rounds of s_i being conducted until round t. Equation (15) shows that the MC platform hopes

Algorithm 3 Balance-Aware Solution (BAS)

Require: α, $\mathcal{S}$, $\mathcal{B}$, ϱ_1, ϱ_2, e_i and c_{ij} for $\forall s_i, s_j \in \mathcal{S}$

1: **Initialization:** Go to the location of each task and conduct it for one round. Update related information.
2: **while** $true$ **do**
3: Select the task with the highest value in (17), $i_{now} = \text{argmax}_{s_i \in \mathcal{S}} \left(\widehat{r}_i(t)/\overline{b}_i(t) - \varrho_1 p_{i_{old}i}(t) + \varrho_2 \cdot Q_i(t) \right)$.
4: Calculate the epoch length for the selected task i_{now}, $l(\tau) = \lceil \tau(E_{i_{now}}(t)+1) - \tau(E_{i_{now}}(t)) \rceil$.
5: Go to the location of i_{now}, conduct it and update related information until the $l(\tau)$-th round when $\mathcal{B}_t \geq 0$, otherwise terminate.
6: **end while**

that for any task s_i, the ratio of the actual execution rounds to all tasks' total rounds can reach its desired value e_i.

Here, we suppose that there exists at least one task selection solution so that the balance constraint can be satisfied. The MC system would tell the worker their required minimum fraction of rounds e_i for $\forall s_i \in \mathcal{S}$ in advance. After considering the balance during the process of task selection, the extended problem becomes more complicated. The other settings are still the same as those in Section II-B.

B. Solution to the Balance Constraint

To solve the balance constraint, we apply the concept of the virtual queue [17] to handle it. More specifically, we first use Q_i to denote the virtual queue for the task s_i and then let $Q_i(t)$ denote the queue length of Q_i in round t. Note that we initialize $Q_i(0) = 0$ for $\forall s_i \in \mathcal{S}$, and then $Q_i(t)$ should be updated as follows:

$$Q_i(t) = \max \Big\{ 0,\ Q_i(t-1) + e_i - \mathbb{I}\{i_{t-1} = i\} \Big\}, \quad (16)$$

where $\mathbb{I}\{\text{true}\} = 1$, $\mathbb{I}\{\text{false}\} = 0$ and i_t represents the index for the sensing task which is conducted in the t-th round.

Based on the virtual queue technique, we design a new task selection algorithm. We still adopt the same definition of epoch to reduce the number of switches among sensing tasks. The difference lies in the computation of the index at the beginning of each epoch. Concretely speaking, the computation of the index will involve the virtual queue length to address the balance constraint. That is, at the beginning of each epoch, the worker would choose the sensing task according to

$$i_{now} = \underset{s_i \in \mathcal{S}}{\text{argmax}} \left(\widehat{r}_i(t)/\overline{b}_i(t) - \varrho_1 \cdot p_{i_{old}i}(t) + \varrho_2 \cdot Q_i(t) \right), \quad (17)$$

where $\widehat{r}_i(t)$ means the UCB-based reward, $\overline{b}_i(t)$ denotes the average resource consumption of s_i, and $p_{i_{old}i}(t)$ indicates the preference penalty of switching tasks. Also, $\varrho_2 > 0$ denotes the controlling parameter, which is used to manage the weight of the virtual queue length in task selection.

Based on the above solution, we propose the balance-aware algorithm (*i.e.*, Balance-Aware Solution for Task Selection, BAS), as shown in Alg. 3. Except for the index calculation at the beginning of each epoch, the extended algorithm has the same structure as PAS. Since the optimization goals of the platform and the worker are to maximize their profits, conflicts are bound to exist. This means that if the worker is constrained by the platform when choosing tasks, his interests will be harmed.

TABLE II
EVALUATION SETTINGS

parameter name	default	range
device budget $\mathcal{B}$	10^6	$10^5 - 10^7$
number of tasks m	100	50-200
parameter α	0.1	0.01-1
parameter ϱ_1	0.1	0.1-1
parameter ϱ_2	0.1	0.1-10
parameter ϵ	0.5	0.1-0.5

VI. PERFORMANCE EVALUATION

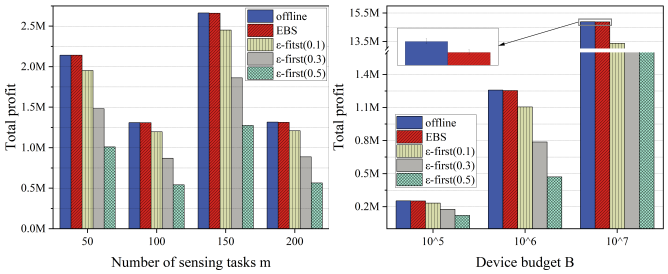
A. Evaluation Methodology

The performance of the proposed algorithms is evaluated based on the real-world traces called Roma-taxi. It contains the GPS coordinates (longitude and latitude) and the corresponding ID of approximately 320 taxi cabs collected over 30 days in Roma, Italy. The positions of the taxi cabs are collected every 7 seconds.

First of all, we stipulate that the tasks are constant in a complete execution process. We select $m \in \{50, 100, 150, 200\}$ GPS coordinates in the central area of Roma and regard them as the locations of m different tasks. Then, we generate the information about the reward, the resource consumed, and the traveling cost. We use the times of all taxi cabs visiting the specific location to generate the task's expected reward (*i.e.*, r_i) and let the distance between the two locations generate the traveling cost. All the reward and resource consumption values are mapped into $(0, 1]$, and traveling cost values are mapped into $[0, 1]$. Since the latitude and longitude values in the trace data can directly generate a distance in $[0, 1]$, we directly use them as the traveling costs. All the values of reward and the resource consumed in each round are generated in the Gaussian Distribution. When generating the values of reward and resource consumed per round, the mean of the Gaussian Distribution is r_i and b_i respectively, while the standard deviation is generated randomly in the range $(0, 1]$.

We present regulations for the parameters of the various algorithms proposed in the paper. The value of α can be chosen from $(0, 1)$, and it was set to be 0.1 by default. Moreover, the values of ϱ_1 in our algorithms are selected from $\{0.1, 0.3, 0.5\}$ and ϱ_2 are selected from $\{0.1, 0.5, 1, 10\}$. Additionally, some other parameters are displayed in Table II.

The algorithms we propose are designed for workers in the MC system and used to solve an optimization problem in an online scenario with multiple constraints. To measure the effectiveness of the algorithms, we use the total profit as a metric to evaluate the optimization performance. In addition, since the PAS considers the traveling cost of switching tasks when selecting tasks, we need to combine the total traveling cost with the total profit as the metric for evaluating it. When selecting tasks, the BAS considers the platform's requirement to balance each task's number of rounds of execution. There-



(a) Total profit vs. m (b) Total profit vs. $\mathcal{B}$

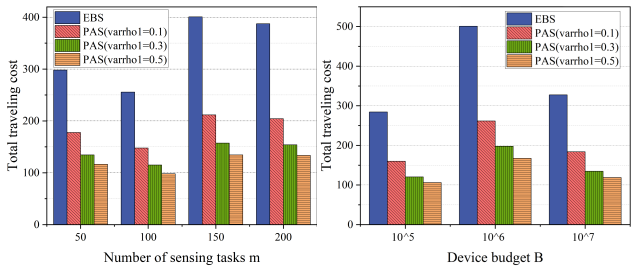
Fig. 3. Total profit of EBS under different parameters m and $\mathcal{B}$.

fore, when showing the algorithm's performance, each task's execution rounds will also become our focus.

Finally, we present the compared algorithms in our simulations. We first implement the offline algorithm based on the known parameters r_i and b_i for $\forall s_i \in \mathcal{S}$. In the offline algorithm, the worker always selects the sensing task with the highest return rate on resources (the ratio of the expected reward and the expected resource consumption) in each round. We then borrow the basic strategy in the famous ϵ-first algorithm for comparison. Concretely speaking, in our model settings, the worker would randomly select one sensing task to conduct when his cumulative consumption of resources does not exceed $\lceil \epsilon \cdot \mathcal{B} \rceil$. In the following rounds, where the worker's remaining device budget is less than $\mathcal{B}$–$\lceil \epsilon \cdot \mathcal{B} \rceil$, he would always select the task with the highest estimated rate of return on resources. In the simulations, the value of ϵ is selected from the set $\{0.1, 0.3, 0.5\}$.

B. Evaluation Results

We first display the evaluation result of EBS based on the real-world traces. To investigate whether the difference in the number of tasks will affect the performance of EBS, we let the total number of tasks in the MC system change when other parameters such as the device budget remain unchanged. We take m from $\{50, 100, 150, 200\}$ and observe the achieved total profit by EBS. The offline algorithm and the ϵ-first algorithm are the comparison algorithms to EBS. When the number of tasks is different, as shown in Fig. 3(a), we find that even though the achieved total profit by EBS cannot surpass the offline algorithm of known task-related information, it is always better than the ϵ-first algorithm. With the increase of m from 50 to 200, the achieved total profit does not have an apparent trend because the reward distributions of the tasks are not intrinsically related. We also examined whether the



(a) Total traveling cost vs. m (b) Total traveling cost vs. $\mathcal{B}$

Fig. 4. Total traveling cost of PAS under different parameters m and $\mathcal{B}$.

TABLE III
COMPARISON OF PAS AND EBS UNDER DIFFERENT TASK NUMBERS

task number m	metrics	PAS(0.1)	PAS(0.3)	PAS(0.5)
50	traveling cost	-40.40%	-54.91%	-60.98%
	total profit	+0.10‰	+0.14‰	+0.16‰
100	traveling cost	-42.24%	-55.02%	-61.68%
	total profit	+0.05‰	+0.07‰	+0.08‰
150	traveling cost	-47.22%	-60.80%	-66.44%
	total profit	+0.08‰	+0.11‰	+0.12‰
200	traveling cost	-47.29%	-60.30%	-65.55%
	total profit	+0.08‰	+0.10‰	+0.13‰

The table shows the comparison results of PAS with different values of ϱ_1 and EBS under different task numbers in specific metrics.

TABLE IV
COMPARISON OF PAS AND EBS UNDER DIFFERENT DEVICE BUDGETS

device budget $\mathcal{B}$	metrics	PAS(0.1)	PAS(0.3)	PAS(0.5)
10^5	traveling cost	-43.74%	-57.67%	-62.73%
	total profit	+0.78‰	+1.00‰	+1.10‰
10^6	traveling cost	-47.74%	-60.52%	-66.58%
	total profit	+0.21‰	+0.25‰	+0.30‰
10^7	traveling cost	-43.87%	-58.89%	-63.71%
	total profit	+0.006‰	+0.010‰	+0.011‰

The table shows the relative differences of PAS with different values of ϱ_1 compared with EBS in specific metrics under different device budgets.

difference in device budget will affect the performance of EBS. Observing Fig. 3(b), we find that when the device budget takes value from $\{10^5, 10^6, 10^7\}$, the total profit obtained after task selection based on EBS is significantly greater than that based on the ϵ-first algorithm. Our observation results show that EBS has significantly better results than the ϵ-first algorithm.

Furthermore, we focus on PAS which considers the worker's preference when selecting tasks. Let ϱ_1 take value from $\{0.1, 0.3, 0.5\}$ in our simulations, reflecting the weight of the traveling-cost-related preference. We also examine the performance of the proposed algorithm from the two dimensions of the number of tasks and the device budget. As shown in Fig. 4(a), when the task number takes different values, compared to using EBS for task selection, using PAS can effectively reduce the total traveling cost. With the increase of ϱ_1, the reduction in total traveling cost is even more significant. Combining Table III, we find that while the total traveling cost is reduced, the total profit will slightly increase. Fig. 4(b) and Table IV show that PAS can effectively reduce the total traveling cost and increase the total profit in the case of different device budgets. And as ϱ_1 increases, the traveling cost and total profit will change more widely. The above experimental results align with our intuition that the greater the weight of the worker's preference, the more likely the worker is to conduct tasks closer to him. Therefore, the total traveling cost is reduced. We believe that PAS has higher usability in scenarios where the traveling cost between tasks is high.

Last, we focus on the evaluation of BAS which involves the balance constraint of the MC platform. Let ϱ_2, which represents the weight of balance, take a value from $\{0.1, 0.5, 1, 10\}$ and then use BAS with different ϱ_2 as the task selection strategy. We observe the number of rounds of each task's execution under different strategies. As shown in Figs. 5-8, with the increase of ϱ_2, the line composed of different tasks' execution rounds gradually tends to be flat. Table V shows the task execution results under BAS with different ϱ_2. We use three metrics to measure the algorithm's effectiveness: the

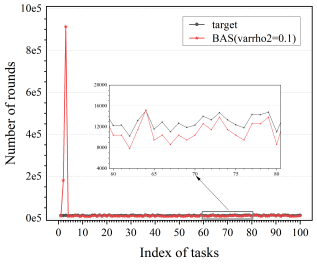


Fig. 5. Execution rounds ($\varrho_2 = 0.1$)

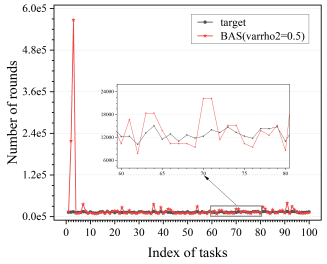


Fig. 6. Execution rounds ($\varrho_2 = 0.5$)

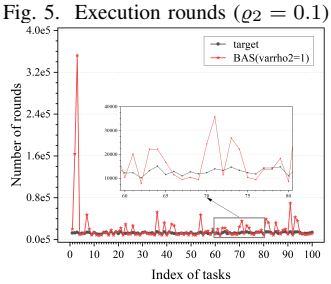


Fig. 7. Execution rounds ($\varrho_2 = 1$)

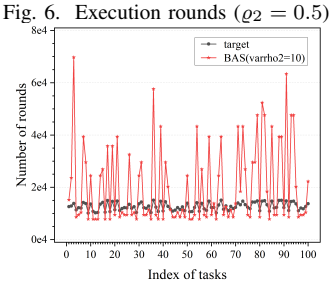


Fig. 8. Execution rounds ($\varrho_2 = 10$)

maximum of the execution rounds, the range of the execution rounds, and the success rate. The meaning of these metrics is shown in Table V. We compare the simulation results based on BAS with the expected result of the MC platform. The information in the target column in Table V indicates that based on the platform's expected minimum execution ratio e_i for $\forall s_i \in \mathcal{S}$, the expected execution rounds of the most executed task is 15083, which is 4913 rounds different from the least executed task. Take BAS with $\varrho_2 = 0.1$ as an example to introduce the meaning of the simulation results. After using it, among all 100 tasks, the actual rounds of the most executed task is 913231, and the difference between the task with the least executed one was 905405. There are 10 tasks that satisfy the platform's requirement for execution rounds, accounting for 10% of the total tasks. Table V shows that when ϱ_2 increases, the difference in the number of rounds of execution between the most and the least executed tasks decreases, which shows that the tasks' execution rounds become more balanced. At the same time, the proportion of tasks whose actual execution rounds reach the values expected by the MC platform shows an upward trend. Fig. 9 shows that, with ϱ_2 increasing, the total profit obtained by the worker after using BAS will decrease significantly. This decrease is intuitive because the increase in ϱ_2 means that the worker has more compromised with the MC platform when choosing tasks.

VII. Related Work

In this paper, we pay attention to the design of a MAB-based strategy to select tasks for a worker when the tasks' information is unknown. At the same time, possible constraints

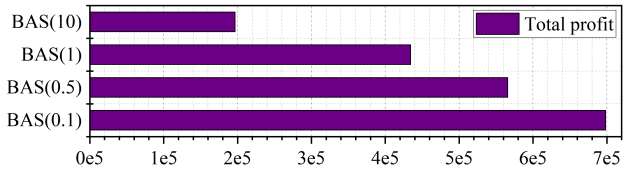


Fig. 9. Total profit of BAS with different values of ϱ_2.

such as the device budget, traveling-cost-related preference, and platform balance requirement are considered. We review related work from two aspects: mobile crowdsensing and multi-armed bandit.

In the current MC researches, most of them [18]–[20] focus on profit maximization from the perspective of the platform while ignoring the entitlement of workers. By contrast, we focus on maximizing total profit from a worker's perspective, which equals the total reward minus the total traveling cost. After investigation, we find that only a few researches [21], [22] study the task assignment problem from the perspective of a worker. For instance, [21] investigates how to maximize the number of location-based tasks conducted by a worker under the deadline constraint and proposes algorithms based on dynamic programming and branch-and-bound strategies. However, it ignores the unknown reward information and constraints in reality, and its practical application is limited. Our work, which is different from the above, makes a meaningful supplement to the research of the MC system.

There are lots of studies [23]–[25] that consider the unknown reward information of sensing tasks. Still, they do not consider the possible constraints in reality such as the device budget, the traveling cost, and so on. For instance, [23] investigates how to maximize the task completion through assigning reliable workers to nearby tasks in spatial crowdsourcing when tasks arrivals are dynamic and worker reliability is unknown. And [24] studies how to draw on logistic-regression techniques from machine learning to learn users' individual preferences from past data and puts forward a new perspective on the payment distribution problem faced by the crowdsensing campaign organizer. Our work considers various constraints from a worker's perspective and our proposed algorithms are more applicable to the actual scene.

We model the unknown task selection problem as a special MAB problem with multiple constraints to solve the dilemma between exploration and exploitation. Because of the existence of the constraints, our unknown task selection problem is more complicated than the traditional MAB problem, and the ideas of the commonly used MAB algorithm [26] cannot be directly applied to solve our problem. Currently, there exist

TABLE V
Task execution results under BAS with different values of ϱ_2

metrics		target	BAS(0)	BAS(0.1)	BAS(0.5)	BAS(1)	BAS(10)
execution rounds	maximum[1]	15083	2517283	913231	567068	352128	69712
	range(max-min)[2]	4913	2517261	905405	559242	344302	61886
success rate[3]		-	1%	10%	39%	43%	44%

[1] The number of execution rounds of the most executed task under a certain task selection strategy.
[2] The difference in the number of execution rounds between the most and the least executed tasks.
[3] The proportion of tasks whose execution rounds reach the expected values of the MC platform.

some studies [27]–[31] about the MAB problem when budget or switching cost exists. Among them, [27] considers the non-stochastic multi-armed bandit problem with switching cost between any pair of actions and gives a tight characterization of the expected minimax regret in this setting. And [28] focuses on the development of MAB in the presence of switching costs and tries to overcome the difficulty of finding the optimal policy in the bandit problem with switching costs. The above studies consider the constraint of either the budget or the switching cost but ignore other constraints which are possible in reality, such as the device budget and platform's balance requirement. Different from the existing studies, we simultaneously consider the constraints of the device, the worker, and the MC platform. Further, we propose some corresponding algorithms to deal with these scenarios.

VIII. Conclusion & Future Work

This paper designs strategy to select tasks from a worker's perspective in the MC system. We consider that the reward information of tasks is unknown, which is in line with our reality. Our problem is modeled as a MAB problem under three constraints: device budget, traveling-cost-related preference, and platform balance requirement. We adopt the idea based on the epoch and design a new task index calculation method for the constraint conditions. Extensive simulations based on real-world traces are conducted to verify the significant performance of our algorithms. In future work, we will try to analyze the worst regret bounds of the proposed algorithms, and meanwhile consider maximizing the total profit for a group of workers while keeping each worker's profit from being harmed since there are always multiple workers in a real MC system (collaboration or competition exists among workers).

Acknowledgement

This research was supported in part by the National Natural Science Foundation of China (Grant No. U20A20182, 62102275, 61873177, 62072322), in part by the NSF of Jiangsu in China under Grant BK20210704, in part by the NSF of the Jiangsu Higher Education Institutions of China under Grant 21KJB520025, and in part by the NSF grants CPS 2128378, CNS 2107014, CNS 2150152, CNS 1824440, CNS 1828363, and CNS 1757533.

References

[1] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.

[2] H. Ma, D. Zhao, and P. Yuan, "Opportunities in mobile crowd sensing," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 29–35, 2014.

[3] B. Guo, Z. Yu, X. Zhou, and D. Zhang, "From participatory sensing to mobile crowd sensing," in *IEEE ERCOM WORKSHOPS*, 2014.

[4] H. Huang, Y. Xin, Y.-E. Sun, and W. Yang, "A truthful double auction mechanism for crowdsensing systems with max-min fairness," in *IEEE WCNC*, 2017.

[5] Y. Du, Y.-E. Sun, H. Huang, L. Huang, H. Xu, Y. Bao, and H. Guo, "Bayesian co-clustering truth discovery for mobile crowd sensing systems," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1045–1057, 2019.

[6] Y. Tong, L. Chen, and C. Shahabi, "Spatial crowdsourcing: Challenges, techniques, and applications," *Proceedings of the VLDB Endowment*, vol. 10, no. 12, pp. 1988–1991, 2017.

[7] P. G. Ipeirotis, F. Provost, and J. Wang, "Quality management on amazon mechanical turk," in *ACM SIGKDD workshop*, 2010.

[8] M. Xiao, J. Wu, L. Huang, Y. Wang, and C. Liu, "Multi-task assignment for crowdsensing in mobile social networks," in *IEEE INFOCOM*, 2015.

[9] M. Karaliopoulos, O. Telelis, and I. Koutsopoulos, "User recruitment for mobile crowdsensing over opportunistic networks," in *IEEE INFOCOM*, 2015.

[10] X. Zhang, Z. Yang, W. Sun, Y. Liu, S. Tang, K. Xing, and X. Mao, "Incentives for mobile crowd sensing: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 54–67, 2015.

[11] L. Wang, D. Zhang, D. Yang, B. Y. Lim, and X. Ma, "Differential location privacy for sparse mobile crowdsensing," in *IEEE ICDM*, 2016.

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

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

[14] G. Gao, J. Wu, M. Xiao, and G. Chen, "Combinatorial multi-armed bandit based unknown worker recruitment in heterogeneous crowdsensing," in *IEEE INFOCOM*, 2020.

[15] X. Gao, S. Chen, and G. Chen, "Mab-based reinforced worker selection framework for budgeted spatial crowdsensing," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2020, early access.

[16] S. Yang, F. Wu, S. Tang, T. Luo, X. Gao, L. Kong, and G. Chen, "Selecting most informative contributors with unknown costs for budgeted crowdsensing," in *IEEE/ACM IWQoS*, 2016.

[17] F. Li, J. Liu, and B. Ji, "Combinatorial sleeping bandits with fairness constraints," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 3, pp. 1799–1813, 2019.

[18] Y. Wu, F. Li, L. Ma, Y. Xie, T. Li, and Y. Wang, "A context-aware multiarmed bandit incentive mechanism for mobile crowd sensing systems," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7648–7658, 2019.

[19] E. Wang, Y. Yang, J. Wu, W. Liu, and X. Wang, "An efficient prediction-based user recruitment for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 17, no. 1, pp. 16–28, 2017.

[20] Y. Liu, B. Guo, Y. Wang, W. Wu, Z. Yu, and D. Zhang, "Taskme: Multi-task allocation in mobile crowd sensing," in *ACM Ubicomp*, 2016.

[21] D. Deng, C. Shahabi, U. Demiryurek, and L. Zhu, "Task selection in spatial crowdsourcing from worker's perspective," *GeoInformatica*, vol. 20, no. 3, pp. 529–568, 2016.

[22] S. Sarker, M. A. Razzaque, M. M. Hassan, A. Almogren, G. Fortino, and M. Zhou, "Optimal selection of crowdsourcing workers balancing their utilities and platform profit," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8602–8614, 2019.

[23] U. ul Hassan and E. Curry, "Efficient task assignment for spatial crowdsourcing: A combinatorial fractional optimization approach with semi-bandit learning," *Expert Systems with Applications*, vol. 58, pp. 36–56, 2016.

[24] M. Karaliopoulos, I. Koutsopoulos, and M. Titsias, "First learn then earn: Optimizing mobile crowdsensing campaigns through data-driven user profiling," in *ACM MobiHoc*, 2016.

[25] S. K. née Müller, C. Tekin, M. van der Schaar, and A. Klein, "Context-aware hierarchical online learning for performance maximization in mobile crowdsourcing," *IEEE/ACM Transactions on Networking*, vol. 26, no. 3, pp. 1334–1347, 2018.

[26] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *arXiv preprint arXiv:1204.5721*, 2012.

[27] T. Koren, R. Livni, and Y. Mansour, "Multi-armed bandits with metric movement costs," *arXiv preprint arXiv:1710.08997*, 2017.

[28] T. Jun, "A survey on the bandit problem with switching costs," *de Economist*, vol. 152, no. 4, pp. 513–541, 2004.

[29] T. Le, C. Szepesvari, and R. Zheng, "Sequential learning for multi-channel wireless network monitoring with channel switching costs," *IEEE Transactions on Signal Processing*, vol. 62, no. 22, pp. 5919–5929, 2014.

[30] G. Gao, H. Huang, M. Xiao, J. Wu, Y.-E. Sun, and S. Zhang, "Auction-based combinatorial multi-armed bandit mechanisms with strategic arms," in *IEEE INFOCOM*, 2021.

[31] R. Agrawal, M. Hegde, D. Teneketzis *et al.*, "Multi-armed bandit problems with multiple plays and switching cost," *Stochastics and Stochastic reports*, vol. 29, no. 4, pp. 437–459, 1990.