TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214
DOI: 10.26599/TST.2024.9010053

Diversity-Based Recruitment in Crowdsensing By Combinatorial Multi-Armed Bandits

Abdalaziz Sawwan* and Jie Wu

Abstract: Mobile crowdsensing (MCS) represents a transformative approach to collecting data from the environment as it utilizes the ubiquity and sensory capabilities of mobile devices with human participants. This paradigm enables scales of data collection critical for applications ranging from environmental monitoring to urban planning. However, the effective harnessing of this distributed data collection capability faces significant challenges. One of the most significant challenges is the variability in the sensing qualities of the participating devices while they are initially unknown and must be learned over time to optimize task assignments. This paper tackles the dual challenges of managing task diversity to mitigate data redundancy and optimizing task assignment amidst the inherent variability of worker performance. We introduce a novel model that dynamically adjusts task weights based on assignment frequency to promote diversity and incorporates a flexible approach to account for the different qualities of task completion, especially in scenarios with overlapping task assignments. Our strategy aims to maximize the overall weighted quality of data collected within the constraints of a predefined budget. Our strategy leverages a combinatorial multi-armed bandit framework with an upper confidence bound approach to guide decision-making. We demonstrate the efficacy of our approach through a combination of regret analysis and simulations grounded in realistic scenarios.

Key words: Diverse allocation, mobile crowdsensing, multi-agent systems, multi-armed bandits, online learning, worker recruitment.

1 Introduction

Recent years have seen a surge in interest in Mobile Crowdsensing (MCS) systems. These systems engage a crowd of mobile users, equipped with their smartphones, to perform various sensing tasks [1–8]. The technology in smartphones enables users to execute a wide array of sensing tasks. Their mobility facilitates extensive area coverage, making these systems ideal for tasks challenging for a single individual, such as noise level collection, traffic data gathering, and water pollution monitoring.

MCS systems offer a cost-effective method for data collection by utilizing the resources of numerous individuals. They hold the potential to gather critical information for diverse applications. Mobile devices allow for flexible and convenient data gathering. In an MCS system, the requester assigns tasks to mobile workers, prioritizing tasks based on their importance. This strategy ensures efficient resource allocation and data collection.

However, a significant challenge arises from the variability in the sensing capabilities of mobile devices used by the workers. This variability affects the quality of data collected and poses a problem for requesters aiming to maximize data quality while minimizing costs. Traditional strategies often assume prior knowledge of these sensing capabilities, which is

• The authors are with The Center for Networked Computing, Temple University, Philadelphia, PA 19122, United States. E-mail addresses: {sawwan,jiewu}@temple.edu.

∗ To whom correspondence should be addressed.

Manuscript received: 2023-Aug-03; revised: 2023-Nov-20; revised: 2024-Feb-28; accepted: 2024-Mar-06.

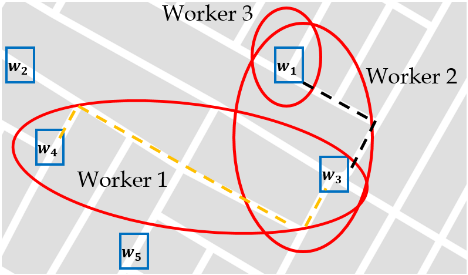


Fig. 1 A basic illustration of one round of the MCS system. Each one of the tasks has a weight w_j that reflects its importance. Each one of the workers has a different sensing quality. Dashed lines represent trajectories for workers.

rarely realistic [9–16]. Moreover, the issue of unknown sensing qualities and the need for diverse task allocation have not been thoroughly addressed.

Addressing this gap, our paper proposes an innovative MCS recruitment strategy that focuses on enhancing task diversity across rounds and learning the sensing qualities of workers' devices over time. We introduce a novel model that dynamically adjusts task assignments to improve data collection quality and efficiency. This approach, integrating a generic expression for handling task overlaps with a classic multi-armed bandit framework, represents a significant advancement in the field of MCS.

In addition, Ul Hassan *et al.* [17] were among the first to study this problem, seeking to maximize the ratio of completed tasks under dynamic task arrival settings. Wu *et al.* [18] developed an algorithm that recruits workers based on the Thompson sampling model, but with the assumption that the sensing quality of workers can vary for each task. Gao *et al.* [19, 20] also considered the case of heterogeneous MCS systems with unknown workers, but their approach used Upper Confidence Bound (UCB) to model the problem, without taking into account task diversity over rounds. It is clear that there is a need for more comprehensive and realistic approaches to solving the problem of recruiting workers in MCS systems, particularly those that can handle the challenges of unknown sensing qualities and diverse task allocation.

In this paper, we consider encouraging choosing diverse tasks over the rounds as we develop an MCS recruitment strategy in which the sensing qualities of workers are being learned. Furthermore, we propose a generic expression with a tunable parameter for overlaps. This expression is seamlessly integrated into the classic multi-armed bandit model and its analysis. Consider the case where the sensing tasks are actually taking videos or pictures of the traffic in a certain city over a period of time where the requester uses the platform to recruit some mobile users who act as the workers. Hence, we have multiple weighted tasks (weight represents importance) that need to be covered in multiple rounds. Repeated coverage of the same task is allowed, with a discounted value. Each worker has multiple possible subsets of tasks, each of which corresponds to a worker's possible daily travel trajectory. Each worker requests a cost with an uncertain quality for task coverage. The objective is to maximize overall utility, defined as a summation of coverage quality multiplied by coverage weight, constrained by the budget. Coverage weight is the weight of each task multiplied by the discounted multiple coverages of the task. Weight values diminish with repetition over the rounds to encourage diversity. Figure 1 shows the basic illustration of the problem. Note that the number of tasks for each worker is not necessarily the same.

In this work, the requester publishes their tasks to the platform. Then, the workers submit their possible sets of tasks to cover alongside the cost values of each one of those possible sets. Afterward, over many rounds, the requester tells each of the workers which of those sets to cover or if they are not recruited, and the workers upload their pictures or videos. The sensing quality of each worker follows an unknown probability distribution to be learned over the rounds. So, our goal is to develop a strategy that maximizes the total weighted quality values constrained by a certain budget.

In our MCS framework, we address several unique challenges to optimize task allocation and worker recruitment. A key challenge is the dynamic adjustment of task weights over successive rounds. Specifically, as a task is covered in one round, its weight value is decreased for subsequent rounds. This approach is designed to prevent the recurrent selection of the same tasks, thereby promoting a diverse range of task coverage. Such a mechanism is vital to avoid data redundancy and ensure a comprehensive understanding of the sensing field. Furthermore, we address the complexity of handling overlaps in task coverage, where tasks may be sensed by multiple workers simultaneously. In this scenario, our model calculates the total completion quality based on both individual and collective worker qualities. This nuanced

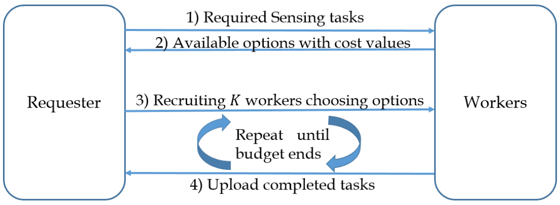

Requester
Workers
1) Required Sensing tasks
2) Available options with cost values
3) Recruiting *K* workers choosing options
Repeat until budget ends
4) Upload completed tasks

Fig. 2 A basic illustration of one round of the MCS system. Each one of the tasks has a weight w_i that reflects its importance. Each one of the workers has a different sensing quality.

treatment of overlaps ensures that the data quality is not compromised despite the concurrent sensing efforts. Additionally, we consider the challenge of learning the unknown sensing qualities of workers over time. This aspect is crucial, as it directly influences the overall quality of the data collected. By integrating these challenges into our Combinatorial Multi-Armed Bandit (CMAB) framework, we propose a novel approach that significantly enhances the effectiveness of mobile crowdsensing systems.

Our new results in this work are summarized as follows:

- We design a new utility model that penalizes repeated coverage in MCS in order to indirectly encourage diverse coverage over the rounds. A novel K-arm CMAB problem is introduced.
- We propose a generic expression with a tunable parameter for overlaps. This expression is seamlessly integrated into the classic multi-armed bandit model and its analysis.
- We run extensive simulations on real-world data in order to compare the performance of our algorithms with existing ones.

The remainder of the paper is organized as follows. In Section 2, a general overview is presented regarding the system and the novelty of it. Section 3 shows the formulation of the problem. Section 4 presents the solution in details. In Section 5, some related works are reviewed. In Section 6, simulation results are presented to evaluate the performance of our solution on real-world traces. Finally, Section 7 gives the conclusion.

2 General Overview

In this section, we formulate the problem after giving a general overview of the MCS system concerned. Now, we consider an MCS arrangement in which there is a crowd of mobile users who are ready to be employed as workers to do a sensing task, say collecting traffic information using the cameras of their mobile devices. On the other hand, there is a requester who operates on a platform in order to communicate with the workers. This requester has a limited budget to do the tasks. Each one of those tasks is fixed in its location and has a specific changing weight value that reflects its relative significance. Sensing the tasks starts with the requester that only publishes the tasks to the workers on the platform.

Afterward, those workers submit a group of subsets of tasks (called options) with the cost they will incur for each one of those options. The first round of sensing starts with the requester recruiting K workers by specifying a certain previously-submitted option for each one of those workers. Then, the workers do their tasks and upload their sensing results to the platform. Then, the second round starts by choosing more K workers and so on until the budget of the requester is spent. Figure 2 shows the steps of the process.

One of the key features of our model is the emphasis on encouraging diversity in the selection of tasks over the various rounds of sensing. To achieve this, we utilize a method of penalizing repetition, which decreases the weight of a task chosen by a certain amount for each round in which it has been previously selected. This approach is designed to promote the diversity of task selection and is demonstrated in Figure 3. In addition, tasks that are covered by multiple workers at the same time, referred to as overlaps, return a value that is dependent on the sensing qualities of all the covering workers. The sensing quality of each worker can range from the maximum sensing quality of the covering workers to the summation of their qualities, as illustrated in Figure 4. The distribution of sensing quality for each worker is not known and is learned over the course of the various rounds of sensing. It is worth noting that we make the assumption that the system is able to maintain the truthfulness, which was studied extensively by Wang *et al.* [21] and Zhang *et al.* [22] for crowdsensing models similar to the one we study in this work.

Furthermore, we assume the security and privacy of the crowdsensing system, and therefore do not need to consider incentives in our model. These warranted assumptions allow us to focus on the task of selecting and allocating tasks to the workers in a manner that promotes diversity and efficient resource utilization.

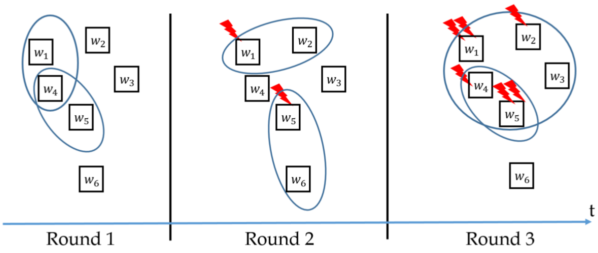


Fig. 3 An illustration of the novel model of diversity in the mobile crowdsensing problem. The red signs include a certain penalty incurred over the utility return from a certain task because of repetition.

There are a variety of frameworks that can be used to quantify the diversity of a chosen set, such as entropy, but our model focuses on penalizing repetition as a means of encouraging diversity.

3 Problem Formulation

In this section, we show the preliminaries needed before formulating the problem.

3.1 Preliminaries

We will use a similar standard notation for mobile crowdsensing systems, [19] is an example. Let the budget of the requester be B, $\mathcal{N}$ represents the set of workers $\{1, \ldots, N\}$, $\mathcal{M}$ represents the set of all tasks $\{1, \ldots, M\}$, and t is the number of the round. The weight of task j at round t is denoted by w_j^t. Without loss of generality, we can start initially with normalized weight values $\Sigma_{j \in \mathcal{M}} w_j^1 = 1$. Normalizing weight values helps us make sense of the relative value for each task in comparison to the total summation of weights of all tasks.

The present system employs a variable, L, to represent the greatest number of options that any worker on the platform has submitted. For any given worker, i, the l-th option among the available options, where $1 \leq l \leq L$, is represented by p_i^l. This variable, p_i^l, comprises a subset of tasks, designated by $\mathcal{M}_i^l$, as well as the cost associated with this particular option, denoted by c_i^l. It is important to note that the subscript i denotes the specific worker in question, while the superscript l refers to the specific option being considered among the available options for that worker.

During each round of the system, the requester is permitted to select a maximum of one option from each worker. Furthermore, it is assumed that the cost of an option, c_i^l, is determined by the number of tasks contained within that option, such that $c_i^l = \epsilon_i f(|\mathcal{M}_i^l|)$. In this equation, $f(.)$ represents a monotonically increasing function, while ϵ_i denotes the cost factor that differentiates the cost values incurred by each individual worker.

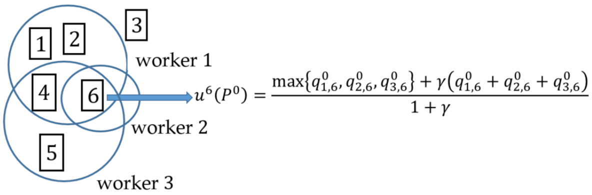

1
2
3
4
5
6
worker 1
worker 2
worker 3
$u^6(P^0) = \frac{\max\{q^0_{1,6}, q^0_{2,6}, q^0_{3,6}\} + \gamma(q^0_{1,6} + q^0_{2,6} + q^0_{3,6})}{1+\gamma}$

Fig. 4 An illustration of having more than one worker covering the same task at the same round 0. The final completion quality of the tasks in overlaps depends on γ.

This model makes the ratio of cost values for two options with the same size from different workers fixed. Furthermore, this cost modeling fixes the cost values of different options of the same size for one worker. It is safe to make that assumption since we are making the cost of an option with a larger number of tasks never cheaper than an option with a smaller number of tasks (although in some special cases this assumption must be relaxed). This modeling of the cost gives the ability to let workers with more advanced devices charge more cost by choosing a proper corresponding cost factor that would reflect how advanced are their smartphones. $\mathcal{O}$ denotes the set of all options of all the workers.

Now, regarding the distributions of the sensing quality of worker i (denoted by Q_i), we assume that it is bounded, and without loss of generality, that it gives a value in $[0, 1]$. That is, $q_{i,j}^t \sim Q_i$, $q_{i,j}^t \in [0, 1]$ where Q_i represents the bounded distribution of the quality values given from worker i, and that $q_{i,j}^t$ is the resulting sample of the distribution when “pulling the arm” and choosing worker i with an option that includes task j at round t.

We make the reasonable assumption that the distributions are independent. we define $q_i = \mathbb{E}[Q_i]$, which is the average sensing quality of worker i, and which is what our strategies try to learn. Having this kind of probability distribution for sensing qualities of workers is realistic as the workers’ sensing qualities have for the same task would be affected by the momentary condition of the worker, their skill, how they use their camera to capture a video or picture of the traffic, etc. Different from classical CMAB models [23, 24], the fact each time an option p_i^l is chosen, the distribution Q_i of sensing quality for worker i is learned many times in the same round. In other words, it will be learned $|\mathcal{M}_i^l|$ times.

One of the most significant innovations in our work is the introduction of a mechanism for dynamically adjusting the weight values w_j of tasks as they are completed over successive rounds. Specifically, as the number of rounds in which task j has been completed, represented by $m_j(t)$, approaches infinity, the weight value approaches a specified fraction, referred to as the diversity ratio and denoted by κ, of the initial weight value ($w_j \rightarrow \kappa w_j^1$ as $m_j(t) \rightarrow \infty$.). This ensures that the weight values of tasks do not become too heavily skewed towards tasks that have been completed less frequently, and instead promotes a more balanced distribution of tasks. This novel approach to task weighting has a number of important implications. By encouraging diversity in the selection of tasks, it helps to prevent the algorithm from becoming too heavily focused on any one particular task, which can lead to suboptimal performance. In addition, by continually adjusting the weight values of tasks as they are completed, it helps to ensure that the algorithm remains responsive to changes in the task environment and can effectively adapt to new circumstances. Overall, the implementation of this task weighting mechanism represents a significant contribution to the field and has the potential to improve the performance of a wide range of algorithms.

The other key contribution of our work is the development of a model for capturing the return on sensing an overlapped task, which is defined as a task that is being sensed by multiple workers simultaneously. This model is based on the assumption that the deterioration in the quality of the task's completion follows an exponential decay function with a rate of $\lambda > 0$. To model the return of sensing an overlapped task, we propose representing the total completion quality of the task as lying within a range between the maximum individual sensing quality of the covering workers and the summation of their sensing qualities. The exact location within this range is determined by a factor called the overlapping factor, denoted by γ, which is a non-negative number. This approach allows us to accurately capture the impact of overlapping on task quality and to account for the diverse ways in which overlapped tasks may be completed. Overall, this novel approach to modeling overlapped tasks represents an important contribution to the field and has the potential to improve the performance of a wide range of algorithms.

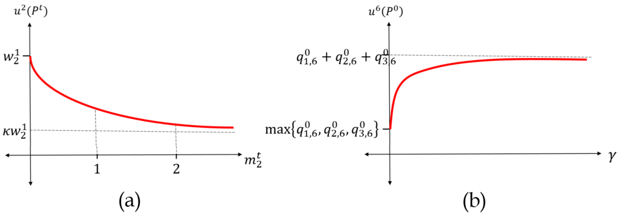


Fig. 5 Plots of the change of utility values in the new model. (a) shows penalizing repetition. (b) shows the overlapping factor.

3.2 The Problem

For this diversity-aware MCS system, our objective is to recruit exactly K workers (choosing the specific option for each worker), in each round in a way that maximized the total weighted sum of completed sensing qualities with the consideration of overlaps over all the rounds. The budget is limited in this regard. Denote the set of all selected options at round t by $\mathcal{O}^t \in \mathcal{O}$ ($p_i^l \in \mathcal{O}^t$ represents the l-th option of worker i selected at round t). To indicate that exactly one option for each worker is selected, $\Sigma_{l=1}^L \mathbb{I}\{p_i^l \in \mathcal{O}^t\} \leq 1$ for all i. Such that $\mathbb{I}\{true\} = 1$ and $\mathbb{I}\{false\} = 1$. Now, the penalizing repetition can be modeled by any monotonically-decreasing function such that $w_j^\infty = \kappa w_j^1$. We opt to choose:

$$w_j^t = ((1-\kappa)e^{-m_j(t)/\lambda} + \kappa)w_j^1 \tag{1}$$

such that:

$$m_j^t = \begin{cases} m_j^{t-1} + 1; & p_i^l \in \mathcal{O}^t \\ m_j^{t-1}; & \text{otherwise} \end{cases} \tag{2}$$

On the other hand, the overlap-aware total quality for a certain task can be modeled as:

$$u^j(\mathcal{O}^t) = \begin{cases} \frac{max\{q_{i,j}^t | p_i^l \in \mathcal{O}^t\} + \gamma(\Sigma_{i|p_i^l \in \mathcal{O}^t} q_{i,j}^t)}{1+\gamma}; & j \in \mathcal{O}^t \\ 0; & \text{otherwise} \end{cases} \tag{3}$$

Both Equations 1 and 3 are plotted in Figure 5. To this end, we can define $u(\mathcal{O}^t)$ to denote the total achieved weighted completion quality of all tasks based on $\mathcal{O}^t$ in round t to be:

$$u(\mathcal{O}^t) = \Sigma_{j \in \mathcal{M}}(w_j^t \times u^j(\mathcal{O}^t)). \tag{4}$$

To see it more clearly, our goal in this work is to determine the set $\{\mathcal{O}^t | \forall t \in [1, \tau(B)]\}$, where $\tau(B)$ is the round at which the budget becomes so small so no more rounds can be done. This set needs to ensure that the expected total weighted completion quality shown in Equation 4 is maximized under the constraints. More formally:

$$\begin{aligned} &\text{maximize}\ \mathbb{E}[\Sigma_t u(\mathcal{O}^t)] && (5)\\ &\text{s. t.}\quad \Sigma_{\forall p_i^l \in \mathcal{O}^t} c_i^l \leq B && (6)\\ &\qquad \Sigma_{\forall l} \mathbb{I}\{p_i^l \in \mathcal{O}^t\} \leq 1 && (7)\\ &\qquad |\mathcal{O}^t| = K \quad \forall t && (8) \end{aligned}$$

In other words, Equation 5 refers to the main objective of maximizing the total expected weighted quality values over the rounds. Equation 6 refers to the budget constraint. Equation 7 refers that at most one option of each worker is chosen every round. finally, the constraint in Equation 8 represents that exactly K workers must be deployed every round.

4 Solution of the Problem

In this section, we introduce the algorithm that solves the novel problem considering both diversity and overlaps.

4.1 Algorithm Overview

The problem at hand is significantly more complex and challenging to analyze compared to previous work in this area, particularly the basic work of Gao *et al.* [20]. This added complexity is due to the inclusion of two additional layers of considerations beyond the fundamental elements of the problem. The first layer pertains to the handling of overlaps or repeated coverage, which is an important factor to consider when determining the optimal allocation of tasks among the workers. This aspect of the problem is described in detail in Equation 3. The second layer involves the modeling of diversity through the penalization of repetition of tasks over time, as outlined in Equation 1. This aspect of the problem is critical for ensuring that the selection of tasks is diverse and well-balanced, rather than overly concentrated on a small number of tasks.

In order to address the issue of having unknown quality distributions for the workers Q_i, we utilize an extended K-CMAB model that is constrained by a budget. This model allows us to consider the trade-off between the cost of recruiting workers and the benefits of their sensing capabilities. Each worker is treated as an arm in this model, and the selection of a worker with an option corresponds to the selection of an arm and a specific option within that arm. The completion quality of an option for a worker is analogous to the reward returned by an arm. By choosing workers and options that provide the highest expected reward given the budget constraints, we can maximize the overall utility of the system. During each round, exactly K workers are recruited, providing the requester with multiple opportunities to learn the sensing quality of each of these workers and adjust their selection strategies accordingly.

In order to do the learning of those quality values, we employ the classical solution of the exploration versus exploitation dichotomy, which is the UCB method [23]. We extend the method to accommodate our application. Afterward, we introduce the estimation function of worker qualities that is based on UCB with the taking of the optimization problem in Equations 5-8. Then, a simple greedy algorithm that chooses K workers each round with the maximum ratio of the estimation quality function to the recruitment cost of the option. Lastly, we show that this greedy algorithm produces a bounded solution in comparison to the optimal one.

4.2 Algorithm Design

Since we assume that, in any round, the requester can force the worker to perform the sensing tasks submitted earlier into the platform as an available option, the worker's quality gets learned by the number of tasks covered by that chosen option $|\mathcal{M}_i^l|$. This raises the need to count how many times the quality of worker i was learned, and how many times a specific option l was chosen.

$$n_i^l(t) = \begin{cases} n_i^l(t) + 1; & p_i^l \in \mathcal{O}^t \\ n_i^l(t-1); & \text{otherwise} \end{cases} \tag{9}$$

$$n_i(t) = \Sigma_{l=1}^{L}((n_i^l(t) - n_i^l(t-1)) \times |\mathcal{M}_i^l|) \tag{10}$$

Equations 9-10 show the record of the times a worker was recruited, and the number of times an option was chosen, at a specific round t.

We now define the average of the measured quality values of worker i at round t. We denote this measured average by $\bar{q}_i(t)$. This average simply equals the following:

$$\bar{q}_i(t) = \begin{cases} \frac{\bar{q}_i(t-1)n_i(t-1) + \Sigma_{j \in \mathcal{M}_i^l} q_{i,j}^t}{n_i(t-1) + |\mathcal{M}_i^l|}; & j \in \mathcal{O}^t \\ \bar{q}_i(t-1); & \text{otherwise} \end{cases} \tag{11}$$

Now, we introduce the classic related UCB equation that balances exploration and exploitation, which is diversity-driven. We use $\hat{q}_i(t)$ to represent this UCB value for quality for worker i. Equation 12 shows it:

$$\hat{q}_i(t) = \bar{q}_i(t) + \sqrt{\frac{(K+1)\log \Sigma_{\forall i} n_i(t)}{n_i(t)}} \tag{12}$$

To introduce the diversity-driven and overlap-aware total completion weighted quality function that is based on the UCB, we define $U(\mathcal{O}^t)$ to denote it at round t in case the set of options $\mathcal{O}^l$ is chosen. That total UCB quality function is shown in Equations 13 -14.

$$U(\mathcal{O}^t) = \Sigma_{\forall j} w_j \hat{u}^j(\mathcal{O}^t) \tag{13}$$

Algorithm 1 Diversity-Based Overlap-Aware Recruitment

```
**Input**: $\mathcal{N}$, $\mathcal{M}$, $\{w_j|\forall j\}$, $B, K, \lambda, \gamma, \kappa$, and $r$.
**Output**: $\{O^1, \ldots, O^{\tau(B)}\}$, and $\Sigma_t u(O^t)$.
1: **While** there is a worker $i_0$ not yet tried **do**
2:     Recruit random $K$ options from $\{p^l_{i_0}|\forall i_0\}$.
3:     Update $B$ and $\{O^1, \ldots\}$. //$B$ to be updated continuously.
4: **While** $\Sigma_{\forall p^l_i \in O^t} c^l_i < B$ **then**
5:     $t = t + 1$, and $O^t = \{\}$.
6:     **While** $|O^t| < K$ **then**
7:         Recruit $\min\{r, K - |O^t|\}$ workers from Equation 15.
8:         Add the recruited workers to $O^t$.
9:     Update $n_i(t), n^l_i(t), \bar{q}_i(t), \hat{q}_i(t), \Sigma_t u(O^t)$, and $B$.
10: **Return** $\{O^1, \ldots, O^{\tau(B)}\}$, and $\Sigma_t u(O^t)$.
```

$$\hat{u}^j(\mathcal{O}^t) = \begin{cases} \frac{max\{\hat{q}^t_{i,j}|p^l_i \in \mathcal{O}^t\} + \gamma(\Sigma_{i|p^l_i \in \mathcal{O}^t} \hat{q}^t_{i,j})}{1+\gamma}; & j \in \mathcal{O}^t \\ 0; & \text{otherwise} \end{cases} \tag{14}$$

This means that our greedy algorithm will simply pick the K workers that maximize the total $U(\mathcal{O}^t)$ value. In more detail, we start by setting a certain accuracy parameter r ($1 \leq r \leq K$) that reflects how accurate $U(\mathcal{O}^t)$ will be evaluated. This parameter determines the maximum number of workers that the algorithm can consider to contribute in an overlap. When we set this parameter to 1, then the algorithm is not overlap-aware anymore. The default value would be 2, and choosing K means that all possible overlaps are considered, but this would require enormous computational power.

When running the algorithm the algorithm will initialize the values of the counters and UCB quality estimations by selecting any option for each worker at the beginning rounds. Then, for each round afterward, the algorithm starts filling up $\mathcal{O}^t$. This is done simply as follows:

(1) Start with empty set $\mathcal{O}^t$.

(2) Find the next $\min\{r, K - |\mathcal{O}^t|\}$ options that would maximize $U_r(\mathcal{O}^t)$ more quickly with total cost.

(3) Repeat the previous step until choosing K workers.

Denoting $U_r(\mathcal{O}^t)$ to represent the value of $U(\mathcal{O}^t)$ evaluated with an accuracy determined by r (considering overlaps with maximum of r contributors).

In other words, our criterion for selection is simply the ratio between the marginal increase in the value of $U_r(\mathcal{O}^t)$ and the cost needed for that increase. More formally, the next options chosen to fill $\mathcal{O}^t$ are:

$$\mathcal{O}^t_{next} = \{p^l_i |\, p^l_i = \underset{\mathcal{P'}^t_{next} \in \mathcal{O}/\mathcal{O}^t}{\arg\max} \frac{U_r(\mathcal{O}^t \cup \mathcal{P'}^t_{next}) - U_r(\mathcal{O}^t)}{\Sigma_{(i,j) \in \mathcal{P'}^t_{next}} c^j_i},$$
$$|\mathcal{P'}^t_{next}| = \min\{r, K - |\mathcal{O}^t|\}\} \tag{15}$$

Notice that in each round, the number of options from different workers chosen is $\min\{r, K - |\mathcal{O}^t|\}$. After depending on Equation 15 to select the K workers at round t, each worker does their specified set of tasks represented by $\mathcal{M}^l_i$, then, all the individual completion qualities $q^t_{i,j}$ are reported to the platform so that the record for each worker is updated. That includes all the recorded averages and UCB values $\bar{q}_i(t)$ and $\hat{q}_i(t)$, alongside the counter values $n_i(t)$ and $n^l_i(t)$, that are used repeatedly all throughout the process. On the other hand, the total accumulated qualities $\Sigma_t u(\mathcal{O}^t)$ is updated. The algorithm keeps iterating until the budget runs out. Algorithm 1 shows the detailed algorithm.

4.3 Algorithm Details and Complexity

The initial lines of the algorithm are designed to initialize various counters and sets that will be used throughout the process. The algorithm then enters a series of rounds, which are executed according to the instructions in lines 3-9. During each round, the algorithm aims to identify the optimal combination of $minr, K - r$ workers based on the criterion outlined in Equation 15. This criterion is meant to ensure that the algorithm is aware of the overlap of up to r workers and seeks to maximize the ratio of marginal gain in total utility to the cost of that gain. The algorithm will continue to run until the budget constraint is reached, at which point it will terminate.

It is important to note that each round t requires workers to report their individual sensing quality values $q^t_{i,j}$, which are used by the platform to update the corresponding counters and measures for each worker. By utilizing this information, the platform is able to keep track of the performance and progress of each worker, allowing it to make more informed decisions about task allocation and resource utilization. Overall, the algorithm is designed to efficiently select the best combination of $minr, K - r$ workers at each round.

Regarding the time complexity of the algorithm, most of the computation is spent in line 7 trying to figure out the solution of Equation 15. Disregarding the initialization, that would take a time of $O(\lceil N/K \rceil)$ to finish. This denomination of line 7 with the loop yield directly to the total time complexity of $O(KL^r N^r M^r)$.

4.4 Bound Performance

If we make the assumption that the distributions of the sensing qualities for each worker Q_i are known and are independent and identically distributed (i.i.d.), the problem can be reduced to a special 0-1 knapsack problem if we set $\gamma = 0$ and $\kappa = 1$. This special case is a well-known NP-hard problem [25], meaning that there is no known polynomial-time algorithm that can solve it optimally. However, our proposed algorithm provides an approximate solution that carefully selects workers according to the criterion outlined in Equation 15. This approximate solution is denoted by $\mathcal{O}^\star$. Despite the fact that it is not a perfect solution, $\mathcal{O}^\star$ still provides satisfactory performance in practice, as it is able to effectively balance the trade-off between maximizing the total utility and minimizing the cost.

This solution always holds that $\hat{u}^j(\mathcal{O}^t) \geq \alpha \times \max_{\mathcal{O}^t \in \mathcal{O}} \hat{u}^j(\mathcal{O}^t)$, such that α is the approximation factor ($0 \leq \alpha \leq 1$). The definition of the overlap-aware $\hat{u}^j(\mathcal{O}^t)$ can be found in Equation 14.

In addition, since we are trying to bound over the $\alpha-$approximation of the optimal solution, we do the $\alpha-$approximation regret analysis [23,26] instead of the traditional one. This means that the regret under a certain budget B becomes defined as in Equations 16-17:

$$R_\alpha(B) = \alpha \times \Sigma_t u(\mathcal{O}^*) - \mathbb{E}[\Sigma_t u(\mathcal{O}^t)] \tag{16}$$

$$R_\alpha(B) \leq \Sigma_t u(\mathcal{O}^\star) - \mathbb{E}[\Sigma_t u(\mathcal{O}^t)] \tag{17}$$

Following the common convention, we denote the optimal workers with superscript $*$, while the α-optimal workers with the $\star$ sign. Now, considering the diversity-based overlap-aware criterion shown in Equation 15, we know beforehand that α for our algorithm can not be less than 0.5 [25, 27]. We now can define both the largest and smallest difference possible in the values of the sensing quality of the workers who are not α-optimal. Equations 18-19 show that:

$$\Delta_{min} = u(\mathcal{O}^\star) - \max_{\mathcal{O}' \neq \mathcal{O}^\star} u(\mathcal{O}') \tag{18}$$

$$\Delta_{max} = u(\mathcal{O}^\star) - \min_{\mathcal{O}' \neq \mathcal{O}^\star} u(\mathcal{O}') \tag{19}$$

Afterward, we need to define a new counter $C_i^l(t)$ for each one of the options. Those counters are just for the options that are chosen when a set of K non-α-optimal workers is chosen. This counter increases by one every time a non-α-optimal set is chosen and the option l from worker i has the least counter value among the other options invoked. That will update this least option as $C_i^l(t) = C_i^l(t-1) + 1$. The expected count of this counter is specifically bounded as in Lemma 1.

Lemma 1 In the system, it holds always that $\mathbb{E}[C_i^l(\tau(B))] \leq \frac{4K^2(K+1)}{(\Delta_{min} \times c_{min})^2} \ln(NM\tau(B)) + 1 + K\pi^2/3$.

Proof The proof of this lemma can, in fact, be derived from previous classic work on CMAB as done in [20]. Starting by defining an indicator that specifies whether $C_i^l(t)$ changes at round t or not.

$$\begin{aligned}
C_i^l(\tau) &= \sum_{\forall t} \mathbb{I}\left\{I_i^l(t) = true\right\} \\
&= x + \sum_{\forall t} \mathbb{I}\left\{I_i^l(t) = true, C_i^l(t) \geq x\right\} \\
&\leq x + \sum_{\forall t} \mathbb{I}\left\{u^j(\mathcal{O}^{t+1}) \geq u^j(\mathcal{O}^\star), C_i^l(t) \geq x\right\} = x + \\
&\sum_{\forall t} \mathbb{I}\left\{\sum_{p_i^l \in \mathcal{O}^{t+1}} \mu_i^l(t+1) \frac{\hat{q}_i(t)}{c_i^l} \geq \sum_{p_i^l \in \mathcal{O}^*} \mu_i^l(\star) \frac{\hat{q}_i(t)}{c_i^l}, C_i^l(t) \geq x\right\}
\end{aligned} \tag{20}$$

such that $\mu_i^l(t+1)$ represents the total product of both the effective number of sensing tasks worker $p_i^l \in \mathcal{O}^{t+1}$ is able to give and the total weight of the corresponding sensing tasks. In other words, it can be shown in Equation 21.

$$\mu_i^l(t+1) = \Sigma_{\forall j} \mathbb{I}\{p_i^l = \arg \max_{p_{i'}^{l'} \in \mathcal{O}^{t+1}} \hat{q}_{i',j}(t+1)\} \times w_j, \tag{21}$$

and we can easily observe that $\mu_i^l(t+1) \leq \sum_{j \in \mathcal{M}_i^l} w_j \leq 1$.

Now, we can use the Chernoff-Hoeffding bound to deduce the fact that x is always lower-bounded as shown in the inequality in Equation 22:

$$\frac{4(K+1)K^2}{(\Delta_{\min} c_{\min})^2}\ln(NM\tau(B)) \leq x \tag{22}$$

To this end, we substitute Equation 22 in Equation 20 with applying the results of the Chernoff-Hoeffding bound from [20,28], we end up with:

$$\begin{aligned} C_i^l(\tau) \leq & \left\lceil \frac{4(K+1)K^2}{(\Delta_{\min} c_{\min})^2}\ln(NM\tau(B)) \right\rceil + \Sigma_{\forall t} 2Kt^{-2} \\ \leq & \frac{4(K+1)K^2}{(\Delta_{\min} c_{\min})^2}\ln(NM\tau(B)) + 1 + \frac{K\pi^2}{3}, \end{aligned} \tag{23}$$

which concludes the proof. ■

Lemma 2 The algorithm terminates at time $\tau(B)$ that is bounded as follows:

$$\tau(B) \leq \frac{2B}{c^\star} + \zeta_4, \tag{24}$$

$$\tau(B) \geq \frac{B}{c^\star} - \zeta_3 - 1 - \frac{\zeta_1 \times \zeta_3}{\zeta_2}\ln(\frac{2B}{c^\star} + \zeta_4). \tag{25}$$

$$\text{where} \begin{cases} c_{min} = \min\{c_i^l | \forall (i,l)\},\ c^\star = \sum_{p_i^l \in \mathcal{O}^\star} c_i^l, \\ c_{max} = \max\{c_i^l | \forall (i,l)\},\ \zeta_1 = \frac{4(K+1)K^2}{(\Delta_{\min} c_{\min})^2}, \\ \zeta_2 = \ln(NM)\zeta_1 + 1 + \frac{K\pi^2}{3}, \\ \zeta_3 = \frac{NL\zeta_2}{c^\star}, \\ \zeta_4 = \frac{2NL}{Kc_{\min}}\left(\zeta_1 \ln\left(\frac{2NL\zeta_1}{Kc_{\min}}\right) - \zeta_1 + \zeta_2\right). \end{cases}$$

Proof First, we present the bound for the α-optimal time $\tau^\star(B)$. Since the set of K workers who are α-optimal is already determined, we can conclude that $\tau^\star(B) = \lfloor B/c^\star \rfloor$. In other words, $\frac{B}{c^\star} - 1 \leq \tau^\star(B) \leq \frac{B}{c^\star}$.

Then, based on the fact that $\forall x > 0,\ x > \ln x + 1$, we get:

$$\ln \tau(B) \leq \frac{Kc_{\min}}{2NL\zeta_1}\tau(B) + \ln\left(\frac{2NL\zeta_1}{Kc_{\min}}\right) - 1 \tag{26}$$

Now, we start from the upper-bound of $\tau(B)$:

$$\begin{aligned} \tau(B) &\leq \tau^\star(B) + \tau(\Sigma_{p_i^l \notin \mathcal{O}^\star} n_i^l(\tau(B))c_{\max}) \\ &\leq \tau^\star(B) + NL/(Kc_{\min})\mathbb{E}[C_i^l(\tau(B))], \end{aligned} \tag{27}$$

Now, to prove the lower bound, we use the same previous notation to use $B^\star$ as the total budget spent on the α-optimal workers. We use B^- to represent the total budget spent on the workers who are not α-optimal. The two values add up to B. From there, we have:

$$\tau(B) = \tau\left(B^\star + B^-\right) \geq \tau\left(B^\star\right) \geq \tau^\star\left(B^\star\right) \tag{28}$$

$$\begin{aligned} &\geq \tau^\star(B - \Sigma_{p_i^l \notin \mathcal{O}_\star} n_i^l(\tau(B))c_{\max}) \\ &\geq \frac{B - NL\mathbb{E}\left[C_i^l(\tau(B))\right]}{c^\star} - 1. \end{aligned} \tag{29}$$

Afterwards, we get:

$$\begin{aligned} &\tau(B) \leq \tau^\star(B) \\ &+ \frac{NL}{Kc_{\min}}(\zeta_1(\frac{Kc_{\min}}{2NL\zeta_1}\tau(B) + \ln(\frac{2NL\zeta_1}{Kc_{\min}}) - 1) + \zeta_2) \\ &\leq \frac{B}{c^\star} + \frac{\tau(B)}{2} + \frac{NL}{Kc_{\min}}(\zeta_1 \ln(\frac{2NL\zeta_1}{Kc_{\min}}) - \zeta_1 + \zeta_2) \\ &\leq \frac{2B}{c^\star} + \frac{2NL}{Kc_{\min}}(\zeta_1 \ln(\frac{2NL\zeta_1}{Kc_{\min}}) - \zeta_1 + \zeta_2) = \frac{2B}{c^\star} + \zeta_4 \end{aligned}$$

Finally, we substitute Equation 29 to get:

$$\begin{aligned} &\tau(B) \geq B/c^\star - NL\zeta_2/c^\star - 1 - NL\zeta_1 \ln(\tau(B))/c^\star \\ &\geq B/c^\star - NL\zeta_2/c^\star - 1 - NL\zeta_1 \ln(2B/c^\star + \zeta_4)/c^\star \\ &= B/c^\star - \zeta_3 - 1 - \ln(2B/c^\star + \zeta_4)\zeta_1\zeta_3/\zeta_2 \end{aligned}$$

■

To this end, we will have the following theorem.

Theorem 1 The upper bound of the $\alpha-$approximation regret of our greedy algorithm is $O\left(NLK^3 \ln B\right)$, in details, $R_\alpha(B) \leq \left(NL\Delta_{\max}\zeta_1 + u^\star\zeta_1\zeta_3/\zeta_2\right)\left(\ln\left(\frac{2B}{c^\star} + \zeta_4\right)\right) + \zeta_5$, where $\zeta_5 = NL\Delta_{\max} + u^\star\left(1/c^\star + \zeta_3 + 1\right)/\kappa$.

Proof Although the asymptotic bound for the regret of our algorithm is the same as the recruitment algorithm without considering the diversity, the exact bound of ours is much different and reflects the diversity factor κ. First, we start with the fact that both Lemma 1 and Lemma 2 yield:

$$\begin{aligned} &R_\alpha(B) \leq \Sigma_{\forall t} u(\mathcal{O}^\star) - \mathbb{E}[\Sigma_{\forall t} u(\mathcal{O}^t)] \\ &\leq \frac{(B+1)u^\star}{c^\star\kappa} - \frac{\tau(B)u^\star}{\kappa} + \tau(B)u^\star - \mathbb{E}[\Sigma_{\forall t} u(\mathcal{O}^t)] \\ &\leq \frac{u^\star}{\kappa}(\frac{B+1}{c^\star} - \tau(B)) + \sum_{i \in \mathcal{N}}\sum_{l=1}^{L} C_i^l(\tau(B))\Delta_{\max} \\ &\leq \frac{u^\star}{\kappa}(\frac{B+1}{c^\star} - (\frac{B}{c^\star} - \zeta_3 - 1 - \frac{\zeta_1\zeta_3}{\zeta_2}\ln(\frac{2B}{c^\star} + \zeta_4)) \\ &+ NL\Delta_{\max}(\zeta_1 \ln(\frac{2B}{c^\star} + \zeta_4) + \zeta_2) \\ &= (NL\Delta_{\max}\zeta_1 + u^\star\zeta_1\zeta_3/\kappa\zeta_2)(\ln(\frac{2B}{c^\star} + \zeta_4)) + \zeta_5 \\ &= O(NLK^3 \ln B). \end{aligned}$$

■

5 Related Work

The development of worker recruitment strategies in MCS systems has attracted many researchers in recent

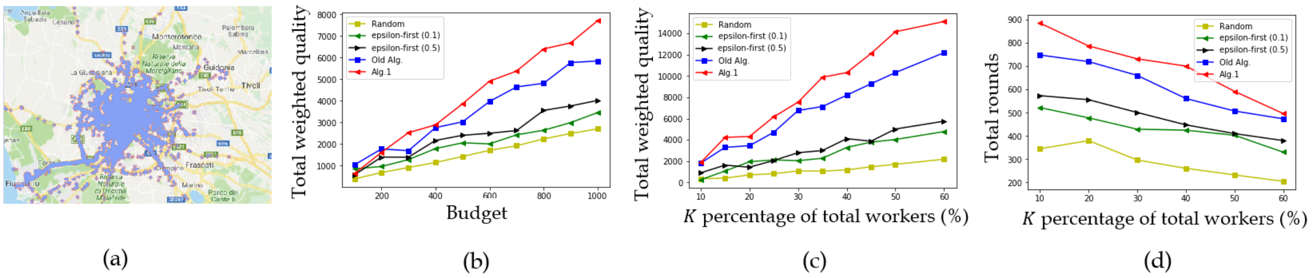


Fig. 6 Simulation results of the basic metrics for default values for parameters. (a) represents the locations in the traces for one of the days. (b) shows the total weighted quality as a function of the total budget. (c) shows the total weighted quality as a function of the percentage of total workers. (d) plots the number of required rounds in terms of K.

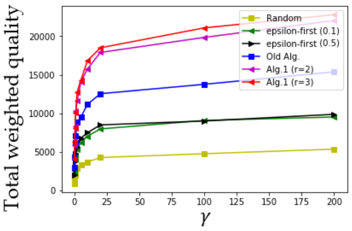


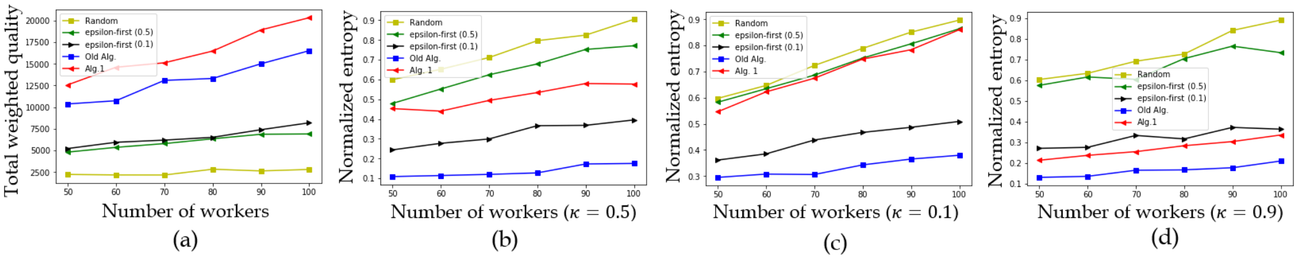
Fig. 7 Simulation results for the total weighted quality with varying γ for overlap-awareness.

years [29–31]. However, most of this work has the typical assumption that the cost values and sensing qualities are not unknown so they try to minimize the cost or maximize the sensing qualities with some constraints. On the other hand, relatively-few work has been done on the case where the sensing values are not known and need to be learned under certain constraints [17, 18, 26, 30, 32–35] and none of them consider the factor of diversity or the general modeling of tasks that are covered by more than one worker in the same round.

Han *et al.* [34] developed an algorithm that maximizes the total reward of sensing tasks in MCS models with the constraint of a limited budget. Yang *et al.* [26] studied the case where the costs of workers are unknown in budgeted MCS systems too. However, both of them either make the assumption that the available options of workers are the same or that the tasks don't differ in terms of significance and don't consider any incentive to make their strategy diversify the set of chosen tasks over the rounds. On the other hand, Karaliopoulos *et al.* [30] studied the problem as a two-dimensional matching problem between the tasks and the workers but without the consideration of different weight values for the tasks.

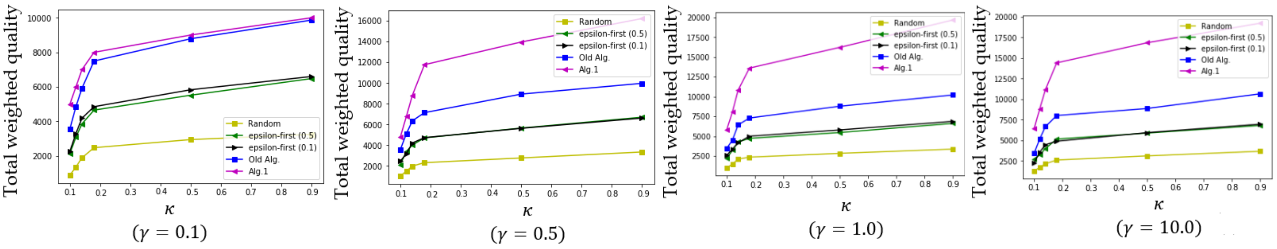
Yang *et al.* [26] and Song and Jin *et al.* [36] present models in mobile crowdsensing that focus on maximizing the overall sensing quality of tasks. Yang *et al.* approach this through selecting the most informative contributors within a budget, while Song and Jin aim to minimize entropy in task selection using a CMAB approach. Both models, while effective in ensuring a diverse range of tasks within a single round, do not explicitly address the challenge of promoting diversity over multiple rounds. Their approaches implicitly encourage a spread of task selection in each round but may lead to repetitive task selections across successive rounds. In contrast, our model introduces a systematic mechanism to actively encourage round-to-round task diversity. By systematically reducing the weight of tasks each time they are selected, our approach ensures a continually evolving pattern of task selection, addressing a gap not explicitly considered in the models by Yang *et al.* and Song and Jin.

Zhou *et al.* [37] developed a UCB-based for the K-CMAB problem but without the consideration of having multiple options for each arm and without the consideration of having overlaps between the tasks that those options model. On the other hand, although Gao *et al.* [20] have developed a novel work that is similar to ours to a certain degree where their CMAB model covers the unknown worker recruitment problem with unknown quality values and flexibility in choosing tasks with different weight values, they do not consider the case where diversity of the chosen tasks over the rounds is encouraged. Furthermore, they consider a single way to model the tasks that are covered by more than one worker at the same time. Our work, in contrast, solves the problem with changing task weight values that reflect a diversity measure while addressing a more general model for overlaps.



(a) (b) (c) (d)

Fig. 8 Simulation results for the effect of the number of workers on the total weighted quality and the normalized entropy.



(γ = 0.1) (γ = 0.5) (γ = 1.0) (γ = 10.0)

(a) (b) (c) (d)

Fig. 9 Simulation results for the total weighted quality with varying our two new parameters κ and γ together.

6 Simulation

In this section, we demonstrate the performance of our solution in comparison with other methods that may be used for the problem.

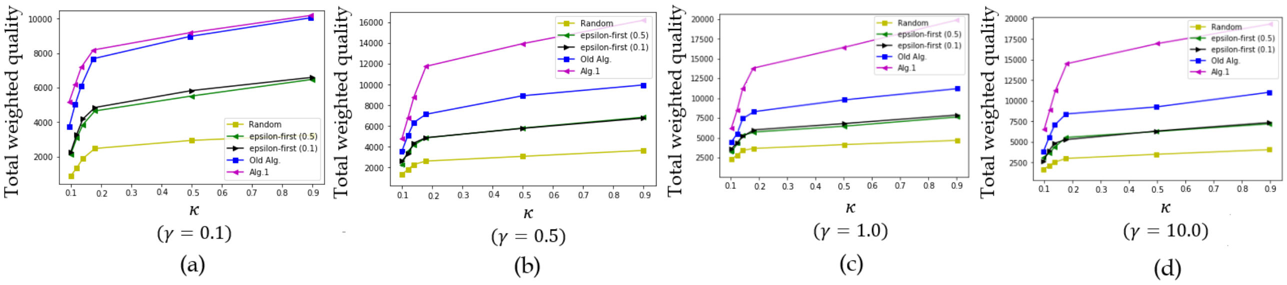
6.1 Experimental Settings

The CRAWDAD dataset [38] is a valuable resource for researchers and practitioners in a variety of fields, as it provides real-world trace data that can be utilized for simulations and analysis. In our particular simulation, we utilize this dataset to evaluate the performance of our algorithm and model. The dataset consists of the locations of approximately 320 taxi cars, which were tracked over a period of 30 days in Rome, Italy. Figure 6 (a) provides a visual representation of the locations of the traces on one day of data collection [20], and we have chosen to focus on a subset of 600 of these locations for the purposes of our simulation. It is important to consider the relevance and reliability of the data being used in any simulation or analysis, and the CRAWDAD dataset has proven to be a reliable and widely-used source of trace data.

In addition, we refine the dataset by removing the taxicabs that rarely visit the considered locations. From those 600 locations, we let M be generated so that 100 to 600 locations are selected. Moreover, in our simulations, we select N taxicabs that perform as workers. N is selected to cover the range $[50, 100]$. We set $M = 300$, $N = 50$, $\kappa = 0.4$, $\gamma = 1$, $\lambda = 5$, $r = 2$, $B = 850$ $K = \lceil \frac{N}{3} \rceil$ to be the default values used whenever they are not mentioned.

Our simulations were conducted on a machine equipped with an Intel(R) Core(TM) i5-7200U CPU operating at 2.50 GHz and supported by 16.0 GB of installed RAM. This hardware configuration provided a robust platform for evaluating the performance of our algorithm, particularly in terms of iteration time. We observed that the iteration time of the algorithm was significantly influenced by the choice of the accuracy parameter r. Specifically, when r was set to 2, the algorithm demonstrated a relatively efficient iteration time, aligning with the computational capabilities of the machine. However, increasing r to 3 resulted in a noticeable increase in the iteration time, reflecting the exponential growth in computational complexity associated with higher r values. Despite this increase, the machine's processing power was sufficient to handle these computations, albeit with extended execution times. This aspect of our experimental setup shows the importance of considering both algorithmic efficiency and hardware capabilities in practical applications, especially for tasks involving complex computations and large datasets.

We set up the choices for each worker by first assigning tasks within a 200 meter radius to each taxicab. Then, we randomly create option subsets



(γ = 0.1)

(a)

(γ = 0.5)

(b)

(γ = 1.0)

(c)

(γ = 10.0)

(d)

Fig. 10 Simulation results for the total weighted quality with varying our two new parameters κ and γ together. The value of $r = 3$.

for each worker, with each subset containing 5 to 15 tasks. For the quality values of each worker's tasks, we choose random means between 0 and 1. We determine the cost of each option based on its size, using a linear relationship. The proportionality factor for this calculation is randomly picked from the range $(0, 1)$. Finally, we distribute the initial task weights w_j^1 across all tasks so that their total equals 1, selecting these weights uniformly at random.

6.2 Algorithm Comparison

In our research, we evaluate our new diversity-based, overlap-aware model against the conventional UCB-based algorithm, dubbed the Old Algorithm, as detailed in [20]. This older method does not account for worker diversity or overlap. We also examine a basic greedy-ϵ strategy. This approach uses a segment of the budget, precisely ϵB, for exploratory random selection of workers and options, allocating the rest for exploiting the top K options. We test with ϵ set to 0.1 and 0.5, noting that ϵ influences the exploration-exploitation budget split. An ϵ of 0.1 leans towards exploitation (90%), favoring stable settings and rapid convergence. In contrast, an ϵ of 0.5 ensures a balance between exploration and exploitation, fitting for the dynamic nature of crowdsensing. This variation in ϵ values facilitates a thorough comparison across different scenarios, showcasing the flexibility of our algorithm in varied environments.

Regarding the quantification of our novel contribution, the diversity is measured here in terms of the *Normalized Entropy*, which is defined as in Equations 30 - 31:

$$\text{Normalized Entropy} = -\Sigma_{\forall j}(p_j \times \log_M p_j), \quad (30)$$

$$p_j = m_j^{\tau(B)} / \Sigma_{\forall j} m_j^{\tau(B)}. \quad (31)$$

The normalized entropy is an excellent measure of diversity as it ranges in $(0, 1]$. Equation 2 provides the formal definition of $m_j^{\tau(B)}$. It is important to note that this measure allows for the quantification of diversity within a given system or dataset.

6.3 Experimental Results

As shown in Figure 6 (b), our proposed algorithm consistently outperforms the old algorithm by approximately 21% on average for various budget values. This trend is also evident when examining the performance of the algorithms for different values of the percentage of total workers K, as depicted in Figure 6 (c). On average, our algorithm outperforms the old algorithm by around 37%. It is worth noting that the diversity-based nature of our algorithm leads to a larger number of rounds being required, as shown in plot (d) of Figure 6. This is due to the fact that our algorithm takes into account the decline in the weight values of tasks that are repeated.

The impact of the overlapping factor γ on the total weighted quality is demonstrated in Figure 7. Higher γ values increase the value of the completion quality of the tasks in overlaps by making it closer to the summation of the completion qualities of those tasks by all workers and farther from the maximum completion qualities of the tasks in overlaps. With larger values of γ, the return of overlaps tends to be significantly higher, as is the case in our used traces, which feature a high number of overlaps. This is why we can see that γ has a dramatic effect on the total weighted quality. Furthermore, as seen in the figure, the impact of changing the value of r is not as pronounced when r is set to values larger then 2. Therefore, it would be more efficient to avoid using high values of r, as increasing this factor significantly increases the time complexity of the algorithm.

One of the most noteworthy results of our analysis can be seen in the plots presented in Figure 8, which illustrate the normalized entropy values of the selected

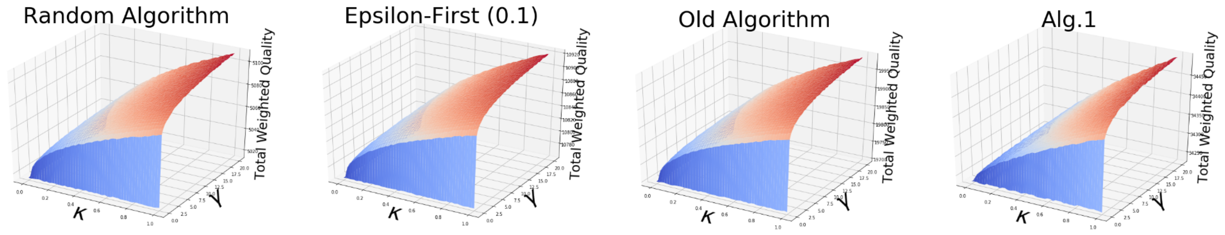


Fig. 11 Exhaustive simulation results for an expanded range of γ and κ.

tasks over all rounds using different algorithms. It is clear from these plots that the purely random algorithm outperforms all other algorithms in terms of diversity, as it consistently chooses a different random set of K options in each round. The ϵ-greedy algorithm with a value of $\epsilon = 0.5$ comes in second place in terms of diversity, as half of the budget is spent on exploring new options by randomly selecting tasks.

Our proposed algorithm secures a broader range of tasks with smaller κ values. That leads to a faster reduction in task weight values. In contrast, the old algorithm lacks in diversity as it fails to recognize the diminishing weight of tasks upon repeated selection. Yet, with larger κ values, the old algorithm's performance aligns more closely with ours. Additionally, Figure 8 illustrates the variation in total weighted quality of tasks as we alter the worker count.

The next set of plots, presented in Figure 9, exhibit the influence of both κ and γ on the total weighted quality values. It can be seen that higher values of κ result in a less severe decline in the weight of the tasks, which allows our proposed algorithm to perform similarly to the old algorithm when γ has a relatively low value. Overall, the results of our simulations align with our theoretical predictions and provide valuable insights into the relationships between the various parameters and the performance of the algorithms.

Following the analysis with $r = 2$, Figure 10 presents a new set of plots that depict the algorithm's performance with the accuracy parameter set to $r = 3$. Remarkably, the results shown in these plots are almost identical to those observed in Figure 9, where $r = 2$. This similarity in outcomes highlights the robustness of our algorithm against changes in the accuracy parameter, particularly in terms of managing task overlaps. Despite the higher computational demand associated with $r = 3$, the algorithm consistently achieves similar utility values, indicating that increases in r do not significantly affect the overall performance. This observation also reinforces the efficiency of our approach in selecting optimal task-worker combinations, effectively minimizing unnecessary overlaps, and thus maintaining performance even as the complexity of the problem increases. These findings further substantiate the practical applicability of our algorithm in real-world scenarios, where computational resources and efficiency are crucial considerations.

Figure 11 concludes the simulation by showing a comparison of the performance of the different algorithms for the wide range of parameters. In evaluating the scalability of our algorithm, the CRAWDAD dataset's inherent constraints must be acknowledged. This dataset determines the boundaries within which scalability can be assessed and Comprises trace data from around 320 taxi vehicles in Rome, Italy. Parameters were meticulously chosen to navigate these limitations effectively—specifying a 200-meter proximity threshold for each worker and setting the task distribution per worker to range from 5 to 15. Such parameters were aimed at stretching the dataset's utility to its fullest, thereby to facilitate a comprehensive examination of our algorithm's scalability. Our refined investigations confirm the algorithm's capacity to scale effectively and to demonstrate consistent and robust performance across the spectrum of the dataset's parameters.

7 Conclusion

Mobile crowdsensing attracted researchers in recent years. In this paper, we considered the problem where the sensing qualities of every worker are not known so they need to be learned during the process. Tasks can get covered by more than one worker in the same round. We introduced a novel model in which the completion quality of tasks covered by more than one worker can range from the maximum individual sensing quality to the total of all sensing qualities of the workers covering it. In addition, we modeled the importance value for each task to decline as the task is sensed more over the rounds in order to encourage a more diverse set of tasks to be covered. We designed a budgeted diversity-driven overlap-aware strategy to recruit the workers in a way that maximizes the total weighted qualities encouraging the diversity of tasks covered over rounds with the consideration of overlaps. We used a combinatorial multi-armed bandit setting to model this problem. We utilized an extended version of the upper confidence bound to develop the recruitment strategy and showed the regret analysis. Finally, we showed a comprehensive simulation applied to real data that showed how our algorithm excels compared to other algorithms. The simulation showed that our algorithm gives 21% more total weighted quality than the existing ones considering this model for different budget values.

Acknowledgment

This research was supported in part by NSF grants SaTC 2310298, CNS 2214940, CPS 2128378, CNS 2107014, and CNS 2150152.

References

[1] Wei, Y., Zhu, Y., Zhu, H., Zhang, Q., & Xue, G. (2015, April). Truthful online double auctions for dynamic mobile crowdsourcing. In 2015 IEEE Conference on Computer Communications (INFOCOM) (pp. 2074-2082). IEEE.

[2] Peng, S., Zhang, B., Yan, Y., & Li, C. (2023). A Multi-Platform Cooperation based Task Assignment Mechanism for Mobile Crowdsensing. IEEE Internet of Things Journal.

[3] Dasari, V. S., Kantarci, B., Pouryazdan, M., Foschini, L., & Girolami, M. (2020). Game theory in mobile crowdsensing: A comprehensive survey. Sensors, 20(7), 2055.

[4] Liu, J., Shen, H., Narman, H. S., Chung, W., & Lin, Z. (2018). A survey of mobile crowdsensing techniques: A critical component for the internet of things. ACM Transactions on Cyber-Physical Systems, 2(3), 1-26.

[5] Zhao, H., Xiao, M., Wu, J., Xu, Y., Huang, H., & Zhang, S. (2020). Differentially private unknown worker recruitment for mobile crowdsensing using multi-armed bandits. IEEE Transactions on Mobile Computing, 20(9), 2779-2794.

[6] Sawwan, A., & Wu, J. (2023, May). A New Framework: Short-Term and Long-Term Returns in Stochastic Multi-Armed Bandit. In 42th IEEE International Conference on Computer Communications (IEEE INFOCOM 2023).

[7] Wu, C., Zhu, Y., Zhang, R., Chen, Y., Wang, F., & Cui, S. (2023). FedAB: Truthful Federated Learning with Auction-based Combinatorial Multi-Armed Bandit. IEEE Internet of Things Journal.

[8] Jin, W., Xiao, M., Li, M., & Guo, L. (2019, April). If you do not care about it, sell it: Trading location privacy in mobile crowd sensing. In IEEE INFOCOM 2019-IEEE Conference on Computer Communications (pp. 1045-1053). IEEE.

[9] Ma, H., Zhao, D., & Yuan, P. (2014). Opportunities in mobile crowd sensing. IEEE Communications Magazine, 52(8), 29-35.

[10] Toward optimal allocation of location dependent tasks in crowdsensing

[11] Pu, L., Chen, X., Xu, J., & Fu, X. (2016, April). Crowdlet: Optimal worker recruitment for self-organized mobile crowdsourcing. In IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications (pp. 1-9). IEEE.

[12] Song, Z., Liu, C. H., Wu, J., Ma, J., & Wang, W. (2014). QoI-aware multitask-oriented dynamic participant selection with budget constraints. IEEE Transactions on Vehicular Technology, 63(9), 4618-4632.

[13] Yang, S., Han, K., Zheng, Z., Tang, S., & Wu, F. (2018, April). Towards personalized task matching in mobile crowdsensing via fine-grained user profiling. In IEEE INFOCOM 2018-IEEE Conference on Computer Communications (pp. 2411-2419). IEEE.

[14] Zhang, Y., Li, P., Zhang, T., Liu, J., Huang, W., & Nie, L. (2023). Dynamic User Recruitment in Edge-aided Mobile Crowdsensing. IEEE Transactions on Vehicular Technology.

[15] Xiao, M., Wu, J., Huang, L., Wang, Y., & Liu, C. (2015, April). Multi-task assignment for crowdsensing in mobile social networks. In 2015 IEEE Conference on Computer Communications (INFOCOM) (pp. 2227-2235). IEEE.

[16] Jin, H., Guo, H., Su, L., Nahrstedt, K., & Wang, X. (2019, April). Dynamic task pricing in multi-requester mobile crowd sensing with markov correlated equilibrium. In IEEE INFOCOM 2019-IEEE Conference on Computer Communications (pp. 1063-1071). IEEE.

[17] ul Hassan, U., & Curry, E. (2016). Efficient task assignment for spatial crowdsourcing: A combinatorial fractional optimization approach with semi-bandit learning. Expert Systems with Applications, 58, 36-56.

[18] Wu, Y., Li, F., Ma, L., Xie, Y., Li, T., & Wang, Y. (2019). A context-aware multiarmed bandit incentive mechanism for mobile crowd sensing systems. IEEE Internet of Things Journal, 6(5), 7648-7658.

[19] Gao, G., Huang, S., Huang, H., Xiao, M., Wu, J., Sun, Y. E., & Zhang, S. (2022). Combination of auction theory and multi-armed bandits: Model, algorithm, and application. IEEE Transactions on Mobile Computing.

[20] Gao, G., Wu, J., Xiao, M., & Chen, G. (2020, July). Combinatorial multi-armed bandit based unknown worker recruitment in heterogeneous crowdsensing. In IEEE INFOCOM 2020-IEEE Conference on Computer Communications (pp. 179-188). IEEE.

[21] Wang, H., Yang, Y., Wang, E., Liu, W., Xu, Y., & Wu, J. (2022). Truthful user recruitment for cooperative crowdsensing task: A combinatorial multi-armed bandit approach. IEEE Transactions on Mobile Computing.

[22] Zhang, C., Zhao, M., Zhu, L., Wu, T., & Liu, X. (2022). Enabling efficient and strong privacy-preserving truth discovery in mobile crowdsensing. IEEE Transactions on Information Forensics and Security, 17, 3569-3581.

[23] Chen, W., Wang, Y., & Yuan, Y. (2013, February). Combinatorial multi-armed bandit: General framework and applications. In International conference on machine learning (pp. 151-159). PMLR.

[24] Gai, Y., Krishnamachari, B., & Jain, R. (2012). Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. IEEE/ACM Transactions on Networking, 20(5), 1466-1478.

[25] Lawler, E. L. (1977, October). Fast approximation algorithms for knapsack problems. In 18th Annual Symposium on Foundations of Computer Science (sfcs 1977) (pp. 206-213). IEEE.

[26] Yang, S., Wu, F., Tang, S., Luo, T., Gao, X., Kong, L., & Chen, G. (2016, June). Selecting most informative contributors with unknown costs for budgeted crowdsensing. In 2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS) (pp. 1-6). IEEE.

[27] Vazirani, V. V. (2001). Approximation algorithms (Vol. 1). Berlin: springer.

[28] Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. Machine learning, 47, 235-256.

[29] Kim, J. W., Edemacu, K., & Jang, B. (2022). Privacy-preserving mechanisms for location privacy in mobile crowdsensing: A survey. Journal of Network and Computer Applications, 200, 103315.

[30] Karaliopoulos, M., Koutsopoulos, I., & Titsias, M. (2016, July). First learn then earn: Optimizing mobile crowdsensing campaigns through data-driven user profiling. In Proceedings of the 17th ACM international symposium on mobile ad hoc networking and computing (pp. 271-280).

[31] Lin, Y., Cai, Z., Wang, X., Hao, F., Wang, L., & Sai, A. M. V. V. (2021). Multi-round incentive mechanism for cold start-enabled mobile crowdsensing. IEEE Transactions on Vehicular Technology, 70(1), 993-1007.

[32] Xiao, M., An, B., Wang, J., Gao, G., Zhang, S., & Wu, J. (2021). Cmab-based reverse auction for unknown worker recruitment in mobile crowdsensing. IEEE Transactions on Mobile Computing, 21(10), 3502-3518.

[33] née Müller, S. K., Tekin, C., van der Schaar, M., & Klein, A. (2018). Context-aware hierarchical online learning for performance maximization in mobile crowdsourcing. IEEE/ACM Transactions on Networking, 26(3), 1334-1347.

[34] Han, K., Zhang, C., & Luo, J. (2015). Taming the uncertainty: Budget limited robust crowdsensing through online learning. Ieee/acm transactions on networking, 24(3), 1462-1475.

[35] Gao, G., Wu, J., Yan, Z., Xiao, M., & Chen, G. (2019, December). Unknown worker recruitment with budget and covering constraints for mobile crowdsensing. In 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS) (pp. 539-547). IEEE.

[36] Song, Y., & Jin, H. (2021, May). Minimizing entropy for crowdsourcing with combinatorial multi-armed bandit. In IEEE INFOCOM 2021-IEEE Conference on Computer Communications (pp. 1-10). IEEE.

[37] Zhou, D., & Tomlin, C. (2018, April). Budget-constrained multi-armed bandits with multiple plays. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 32, No. 1).

[38] Bracciale, L., Bonola, M., Loreti, P., Bianchi, G., Amici, R., Rabuffi, A., CRAWDAD dataset roma/taxi (v. 2014-07-17), downloaded from https://crawdad.org/roma/taxi/20140717, https://doi.org/10.15783/C7QC7M, Jul 2014.

Abdalaziz Sawwan is a fourth-year Ph.D. student in Computer and Information Sciences at Temple University. Sawwan received his bachelor's degree in Electrical Engineering from the University of Jordan in 2020 and his master's degree from Temple University in 2023. His current research interests include learning theory, multi-armed bandits, reinforcement learning, and wireless networks.

Jie Wu is the Director of the Center for Networked Computing and Laura H. Carnell professor at Temple University. His current research interests include mobile computing and wireless networks, cloud computing, and network trust and security. Dr. Wu regularly published in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Service Computing, IEEE/ACM Transactions on Networking, and Journal of Computer Science and Technology. Dr. Wu is/was general chair/co-chair for IEEE DCOSS'09, IEEE ICDCS'13, ICPP'16, IEEE CNS'16, WiOpt'21, ICDCN'22, IEEE IPDPS'23, and ACM MobiHoc'23 as well as program chair/cochair for IEEE MASS'04, IEEE INFOCOM'11, CCF CNCC'13, and ICCCN'20. Dr. Wu is a Fellow of the AAAS and a Fellow of the IEEE.