Deadline-Sensitive User Recruitment for Mobile Crowdsensing with Probabilistic Collaboration

Mingjun Xiao*†, Jie Wu†, He Huang‡, Liusheng Huang*, and Chang Hu*
*School of Computer Science and Technology / Suzhou Institute for Advanced Study,
University of Science and Technology of China
†Department of Computer and Information Sciences, Temple University
‡School of Computer Science and Technology, Soochow University, China

***Abstract*—Mobile crowdsensing is a new paradigm in which a group of mobile users exploit their smart devices to cooperatively perform a large-scale sensing job over urban environments. In this paper, we focus on the Deadline-sensitive User Recruitment (DUR) problem for probabilistically collaborative mobile crowdsensing. Unlike previous works, mobile users in this problem perform sensing tasks with probabilities, and multiple users might be recruited to cooperatively perform a common task, ensuring that the expected completion time is no larger than a deadline. Owing to such a probabilistic collaboration, DUR can be formalized as a non-trivial set cover problem with non-linear programming constraints and an optimization objective of real function. We first prove that the DUR problem is NP-hard. Then, we propose a greedy DUR algorithm, called *g*DUR, to solve this problem. Next, we prove that the *g*DUR algorithm can achieve a logarithmic approximation ratio. Furthermore, we extend the problem to a more complex case where sensing duration is taken into consideration, and we propose a sensing-duration-aware user recruitment algorithm, called *d*DUR. Finally, we validate the performance of the proposed algorithms through extensive simulations, based on a real mobile social network trace and a synthetic trace.**

***Index Terms*—Crowdsensing, mobile social network, probabilistic collaboration, user recruitment**

I. Introduction

In recent years, there has been an explosive proliferation of smartphones. These smartphones have generally been equipped with multi-core processors, gigabytes of memory, and diverse sensors, so that they can be seen as powerful mobile sensors. Thanks to this, a new sensing paradigm, called *mobile crowdsensing*, is proposed [4]. Roughly speaking, mobile crowdsensing refers to a group of mobile users being coordinated to perform a large-scale sensing job over urban environments through their smartphones. Since mobile crowdsensing can perform sensing jobs that individual users cannot cope with, it has stimulated many applications, such as urban WiFi characterization, traffic information mapping, wireless indoor localization, and so on, attracting much attention [3], [5], [12], [18], [25].

By far, there has been much research on mobile crowdsensing, including platform design, user recruitment or task allocation algorithms, incentive mechanisms, and so on [1], [2], [6]–[8], [10], [14], [15], [19], [21]–[24]. Among them, user recruitment or task allocation is one of the most important topics [2], [6], [7], [10], [15], [21], [24]. In this paper, we focus on the Deadline-sensitive User Recruitment (DUR) problem for probabilistically collaborative mobile crowdsensing. More specifically, a requester wants to collect some sensing data from many points of interest (PoIs) in an urban area, just like [2], [6], [7], [10]. Then, it publishes a crowdsensing request to some mobile users via mobile social networks. These mobile users move around in the urban area every day. Each user might pass by (i.e., *cover*) some PoIs frequently, so that it can collect the related sensing data with some probabilities, as shown in Fig. 1. If a mobile user participates in crowdsensing, it will charge a cost from the requester as the reward. The crowdsensing job is expected to be accomplished before a given deadline. Our main concern is determining which users should be recruited, so that the requester can minimize the total cost, while ensuring that the expected completion time of the crowdsensing is no larger than the given deadline.

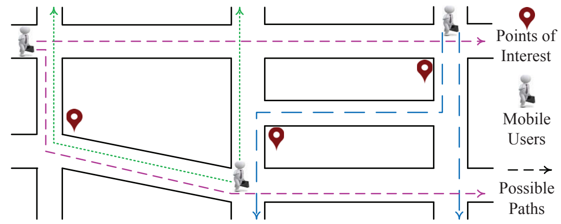


Fig. 1. An example of mobile crowdsensing with probabilistic collaboration: three mobile users move around in an urban area, each of which passes by some points of interest (subject to its mobility) and collects sensing data every day with some probabilities, until the data is collected successfully.

Our DUR problem for probabilistically collaborative mobile crowdsensing in this paper differs from existing user recruitment problems. Existing works mainly focus on deterministic mobile crowdsensing, in which the trajectory of each user is known and deterministic (e.g., [7], [10], [15]), or each user will determine a route (e.g., [2], [6]) for performing sensing tasks; when the trajectory or route covers a PoI, the user can successfully perform its sensing tasks alone, without any cooperation among users. In contrast, our problem is based on the observation that mobile users in real traces do not always move along a fixed path, showing the characteristic of non-deterministic mobility [10], [13], [21]. That is to say, it is probabilistic that mobile users perform sensing tasks. Hence, multiple users need be recruited to cooperatively perform a common task, in order to improve the probability of success. Due to this probabilistic collaboration, our problem can be formalized as a non-trivial set cover problem with non-linear programming constraints and an optimization objective of real function. The methodology in existing works [2], [6], [7], [10], [15] cannot deal with such a problem [11].

In this paper, we carefully design a utility function, based on which we turn our DUR problem into a Minimum Submodular Cover with Submodular Cost (MSC/SC) problem, and adopt the greedy strategy to solve this problem. Furthermore, we also extend the problem and the solution to a more complex case. More specifically, the major contributions include:

1) We introduce a DUR problem for mobile crowdsensing with probabilistic collaboration. Unlike existing works, mobile users in this problem perform sensing tasks with probabilities, and multiple users need to cooperatively perform a common task, resulting in a non-trivial set cover problem with non-linear programming constraints.
2) We prove the NP-hardness of the DUR problem. Moreover, we propose a utility function, and prove it to be submodular. Owing to this utility function, we turn the DUR problem into an MSC/SC problem to be solved by using a greedy approximation algorithm, called *g*DUR. In addition, we derive the logarithmic approximation ratio of the *g*DUR algorithm.
3) We extend the DUR problem to the case, where sensing duration is taken into consideration. To solve this problem, we also propose a submodular utility function, and based on which, we design a sensing-duration-aware user recruitment algorithm, called *d*DUR. Moreover, we analyze the corresponding approximation ratio.
4) We conduct extensive simulations on a real trace and a synthetic trace to prove the significant performances of the proposed algorithms.

The remainder of the paper is organized as follows. We first review related works in Section II. Then, we introduce the model, and the problem in Section III. The *g*DUR and *d*DUR algorithms are proposed in Sections IV and V, respectively. In Section VI, we evaluate the performances of our algorithms through extensive simulations. We conclude the paper in Section VII. *Some complex proofs are moved to the Appendix.*

II. Related Works

There has been much research on the user recruitment or task allocation problem of mobile crowdsensing [2], [6], [7], [10], [21], [24]. Most of these works focus on the deterministic mobile crowdsensing. For example, M. Cheung et al. in [2] formulate a movement-related task allocation problem as a task selection game, and propose a distributed algorithm for each user to select its task and determine its movement. Z. He et al. in [7] propose a greedy approximation algorithm and a genetic algorithm for the user recruitment problem of crowdsensing in vehicular networks, where future trajectories of users are taken into account. S. He et al. in [6] considered the task allocation problem with the constraint of time budgets. In these works, the trajectory of each user is known and deterministic (e.g., [7]), or each user will determine a route (e.g., [2], [6]) for performing sensing tasks, and when the trajectory or route covers a PoI, the user can successfully perform its sensing tasks alone. There is no cooperation among users, different from our probabilistically collaborative DUR problem.

Among the existing works, only [10] partially discussed a special non-deterministic mobile crowdsensing. In this work, M. Karaliopoulos et al. studied the problem of recruiting the users whose paths can cover some PoIs with a minimum cost. The authors first consider a deterministic mobility scenario, where each user has a fixed path, and they formulate the user recruitment problem as a trivial set cover problem with a submodular objective function to be solved. Then, the authors also discuss the non-deterministic scenario, in which users' paths might cover PoIs with some probabilities. In this scenario, the authors let the additive sum of probabilities of each PoI being covered by multiple paths be no less than 1. For example, if a PoI is covered by two paths with the probabilities p_1 and p_2, then they must be subject to $p_1+p_2 \geq 1$. Since the constraint based on the additive probability is a linear programming constraint, the problem in [10] is still a trivial set cover problem. In contrast, our problem is a probabilistic set cover problem with non-linear programming constraints and an optimization objective of real function. The methodology in [10], which is only suitable for the set cover problems with linear programming constraints, cannot deal with our problem, according to the corresponding theories in [10], [11]. In the following, we will design a novel utility function, based on which we can turn it into an MSC/SC problem to be solved.

Additionally, Q. Zhao et al. in [24] and M. Xiao et al. in [21] studied the task allocation issues by formulating them as online scheduling problems, also different from our problem.

III. Model & Problem

A. Model

We consider a mobile crowdsensing, in which a *requester* wants to collect some sensing data from many PoIs in an urban area. The crowdsensing job is assumed to have been divided into many sensing tasks according to the PoIs, denoted by $\mathcal{S}=\{s_1, s_2, \cdots, s_m\}$, where each task s_j $(1 \leq j \leq m)$ is related to a specific PoI. On the other hand, there are many mobile social network users, moving around in the urban area every day. Many real traces demonstrate that mobile social network users will periodically visit some locations that they prefer with probabilities [9], [16], [20]. For example, mobile users might go to their offices, homes, shopping malls, or other places every day. As a result, each user might periodically pass by (i.e., *cover*) one or more specific PoIs with probabilities. Therefore, these users can be recruited by the requester to perform the corresponding sensing tasks. In this paper, we only discuss the users who are willing to participate in the crowdsensing, denoted by $\mathcal{U}=\{u_1, ..., u_n\}$. The users who can perform task s_j are denoted as $\mathcal{U}_j$, and the tasks that user u_i $(1 \leq i \leq n)$ can deal with are denoted as $\mathcal{S}_i$. Moreover, according to the users' periodic mobile behaviors, time is divided into many equal-length sensing cycles, denoted by τ. For instance, a sensing cycle might be a day, or several hours, etc. Then, the detailed crowdsensing is conducted as follows.

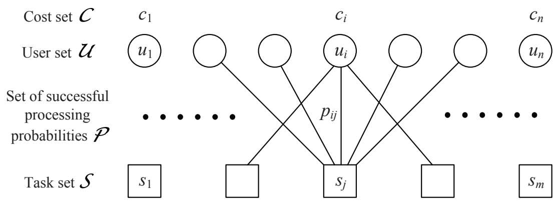


Fig. 2. A probabilistically collaborative mobile crowdsensing can be simply described by $\langle \mathcal{U}, \mathcal{S}, \mathcal{P}, \mathcal{C} \rangle$, where $\mathcal{U}$ is the user set, $\mathcal{S}$ is the task set, $\mathcal{P}$ is the set of probabilities of users' performing tasks in each sensing cycle, and $\mathcal{C}$ is the set of costs claimed by users.

First, the requester publishes all sensing tasks in $\mathcal{S}$ to the users in $\mathcal{U}$ via mobile social networks. Then, each user u_i can determine the tasks that it can perform, i.e., $\mathcal{S}_i$. Meanwhile, user u_i can also determine the probability of performing every task $s_j \in \mathcal{S}_i$ in each sensing cycle, called *the processing probability* and denoted by p_{ij}. Actually, many mobile phones provide the functionality to record the trajectories of users. Through this functionality, u_i can derive the frequency of visiting the PoI related to task s_j, which can be used to estimate p_{ij}. For example, if u_i has recorded its trajectories of 5 sensing cycles, among which it has visited the PoI in 3 sensing cycles, it will set $p_{ij} = 0.6$. Here, if user u_i visited multiple times in a same sensing cycle, it will still be counted as one time, since we only concern whether the user can perform the task in this sensing cycle.

Second, after having derived each sensing probability p_{ij}, user u_i will tell the requester which tasks it can deal with and the related probabilities of performing these tasks via the mobile social networks. At the same time, user u_i will also tell the requester that it will charge a cost from the requester as the reward for participating in crowdsensing. We assume that there is a negotiation between the requester and the user, during which they can determine the cost, denoted by c_i. In fact, many mechanism can be used to produce the cost and ensure the truthfulness, such as [19], [23]. In this paper, we will not discuss the detailed negotiation mechanism.

Finally, according to the responses from the users in $\mathcal{U}$, the requester makes the decision to recruit some users for performing the sensing tasks in $\mathcal{S}$. If a user u_i is recruited by the requester, it will perform each task in $\mathcal{S}_i$, when passing by the corresponding PoI, until it is told that the task has been completed by other users. In order to improve the probability of success, multiple users might be recruited to perform a common task. Once having completed a task, each user will send the results back to the requester via the mobile social networks.

Let $\mathcal{P} = \{p_{ij} | u_i \in \mathcal{U}, s_j \in \mathcal{S}_i\}$ and $\mathcal{C} = \{c_i | u_i \in \mathcal{U}\}$. Then, we can simply use a four tuple $\langle \mathcal{U}, \mathcal{S}, \mathcal{P}, \mathcal{C} \rangle$ to describe the above mobile crowdsensing with probabilistic collaboration, as shown in Fig. 2.

TABLE I
DESCRIPTION OF MAJOR NOTATIONS.

Variable	Description
$\mathcal{U}, \mathcal{S}, \mathcal{C}, \mathcal{P}$	the sets of all users, tasks, costs, and probabilities of users performing tasks.
$u_i, s_j, c_i, p_{i,j}$	the i-th user, the j-th task, the cost that user u_i charges from the requester, and the probability of user u_i performing task s_j.
τ, $\mathcal{T}$	the sensing cycle, and the deadline.
$\mathcal{U}_j$	the set of users who can process task s_j.
$\mathcal{S}_i$	the set of tasks that user u_i can deal with.
Φ	the user recruitment solution, i.e., the set of users that the requester recruits.
ρ_j^{Φ}	the joint probability of task s_j being processed successfully by users in Φ at a sensing cycle (Eq. 1).
d_{ij}	the average duration of user u_i performing task s_j in a sensing cycle.
σ_j^{Φ}	the total expected duration of task s_j being sensed by users in Φ at a sensing cycle (Eq. 7).

B. Problem

We focus on the DUR problem in the above mobile crowdsensing, i.e., which users in $\mathcal{U}$ should be recruited by the requester to perform the tasks in $\mathcal{S}$, so that it can minimize the total cost, while ensuring that the expected completion time of the crowdsensing is no larger than a given *deadline* $\mathcal{T}$. As multiple users might be recruited to perform a task, the probability of a task being processed is actually a joint probability, called *joint processing probability*. We use the set Φ to denote a user recruitment solution, where $u_i \in \Phi$ indicates that user u_i is recruited. Moreover, we denote the joint processing probability of task s_j in each sensing cycle as ρ_j^{Φ}. Then, ρ_j^{Φ} can be calculated as follows:

$$\rho_j^{\Phi} = 1 - \prod_{u_i \in \Phi} (1 - p_{ij}) \tag{1}$$

Based on the joint processing probability of task s_j in each sensing cycle, we can get the corresponding expected completion time, i.e., $\frac{\tau}{\rho_j^{\Phi}}$. Further, the DUR problem can be formalized as follows:

$$Minimize: \quad C(\Phi) = \textstyle\sum_{u_i \in \Phi} c_i \tag{2}$$

$$Subject\ to: \quad \Phi \subseteq \mathcal{U} \tag{3}$$

$$\frac{\tau}{\rho_j^{\Phi}} \leq \mathcal{T},\ 1 \leq j \leq m \tag{4}$$

Here, Eq.4 indicates that the expected completion time of each task is no larger than the given deadline $\mathcal{T}$. In other words, the joint processing probability of each task s_j is no less than $\frac{\tau}{\mathcal{T}}$, i.e., $\rho_j^{\Phi} = 1 - \prod_{u_i \in \Phi}(1 - p_{ij}) \geq \frac{\tau}{\mathcal{T}}$. It is this constraint that makes our user recruitment become a set cover problem with non-linear programming constraints, different from the trivial set cover problems in existing works [2], [6], [7], [10], [15].

In this paper, we assume that the DUR problem has at least a feasible solution. That is to say, each task s_j can be performed before the deadline, i.e., $\rho_j^{\mathcal{U}} = 1 - \prod_{u_i \in \mathcal{U}}(1 - p_{ij}) \geq \frac{\tau}{\mathcal{T}}$. If there is no feasible solution for the problem, the requester can have the problem be solvable by expanding the user set $\mathcal{U}$ or prolonging the deadline $\mathcal{T}$. Moreover, for simplicity, $\mathcal{T}$ is assumed to be an integral multiple of τ; otherwise, we can use $\rho_j^{\Phi} \geq 1/\lfloor \frac{\mathcal{T}}{\tau} \rfloor$ to replace $\frac{\tau}{\rho_j^{\Phi}} \leq \mathcal{T}$ in Eq.4, where $\lfloor \frac{\mathcal{T}}{\tau} \rfloor$ is the floor of $\frac{\mathcal{T}}{\tau}$. The following sections will show that our algorithms can still work well in this case.

Algorithm 1 The gDUR Algorithm

Require: $\mathcal{U}, \mathcal{S}, \mathcal{P}, \mathcal{C}, \tau, \mathcal{T}$
Ensure: Φ
1: $\Phi = \emptyset$;
2: **while** $f(\Phi) < \frac{m\tau\theta}{\mathcal{T}}$ **do**
3: Select a user $u_i \in \mathcal{U} \backslash \Phi$ to maximize $\frac{f(\Phi \cup \{u_i\}) - f(\Phi)}{c_i}$;
4: $\Phi = \Phi \cup \{u_i\}$;
5: **return** Φ;

In addition, we extend the DUR problem to a more practical case. We leave the extended DUR problem to be discussed in Section V, for the integrity of description. Here, we also list the main notations in Table I.

IV. Deadline-Sensitive User Recruitment

In this section, we first analyze the complexity of the DUR problem. Then, we propose the gDUR algorithm based on a utility function, followed by an example. Finally, we prove the correctness and approximation ratio of the algorithm.

A. Problem Hardness Analysis

Before the solution, we first prove that the DUR problem is NP-hard, as shown in the following theorem.

***Theorem** 1:* The DUR problem is NP-hard.

Proof: We consider a special case of the DUR problem: given a probabilistically collaborative mobile crowdsensing $\langle \mathcal{U}, \mathcal{S}, \mathcal{P}, \mathcal{C} \rangle$ and a deadline $\mathcal{T}$, where $\mathcal{P} = \{p_{ij} = p \mid 0 < p \leq 1, u_i \in \mathcal{U}, s_j \in \mathcal{S}_i\}$, $\mathcal{C} = \{c_i = 1 \mid u_i \in \mathcal{U}\}$, and $\mathcal{T} = \frac{\tau}{p}$, determine a set $\Phi \subseteq \mathcal{U}$, such that the requester can minimize $C(\Phi) = \sum_{u_i \in \Phi} c_i = |\Phi|$, while the expected processing time of each task is no larger than $\mathcal{T}$. Actually, this special DUR problem is to select the minimum number of users from $\mathcal{U}$ who can process all tasks in $\mathcal{S}$. When we replace each u_i in $\mathcal{U}$ by using $\mathcal{S}_i$ ($\subseteq \mathcal{S}$), i.e., the set of tasks that u_i can process, this problem can be seen as a set cover problem, a well known NP-hard problem: given a task set $\mathcal{S}$, a collection of subset $\{\mathcal{S}_i | 1 \leq i \leq n\}$, find a minimum size of subcollection of $\{\mathcal{S}_i | 1 \leq i \leq n\}$ that covers all tasks in $\mathcal{S}$. That is to say, the special DUR problem is NP-hard. Consequently, the general DUR problem is also at least NP-hard. The theorem holds. ■

B. The gDUR Algorithm

Since DUR is NP-hard, we propose a greedy algorithm to solve it. The greedy criterion is that the user who has the largest probability to process the most tasks with the least cost will be recruited and added into the set Φ first, which is based on the following utility function:

***Definition** 1: Utility function* $f(\Phi)$ indicates the utility about the total probability of the users in set Φ processing all tasks in $\mathcal{S}$ before the deadline, defined as follows:

$$f(\Phi) = \theta \sum_{j=1}^{m} \min\{\rho_j^{\Phi}, \frac{\tau}{\mathcal{T}}\}, \tag{5}$$

where $\theta = \max\{\theta_1, \theta_2\}$ is a constant related to the approximation factor of the gDUR algorithm, in which $\theta_1 = \frac{\mathcal{T}\sum_{i=1}^{n} c_i}{m\tau}$, and $\theta_2 = \max\{\frac{c_i | 1 \leq i \leq n}{\frac{\tau}{\mathcal{T}} - \rho_j^{\Phi} | 1 \leq j \leq m, \rho_j^{\Phi} < \frac{\tau}{\mathcal{T}}, \Phi \subset \mathcal{U}}\}$.

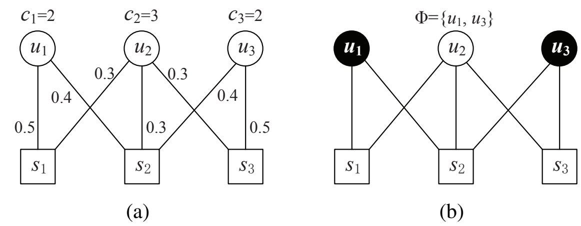


Fig. 3. Illustration of the gDUR algorithm.

By using this defined utility function, we can turn our problem into an MSC/SC problem. We will demonstrate this in the following subsections. Here, we only present the gDUR algorithm based on this utility function, as shown in Algorithm 1. The gDUR algorithm starts from an empty user set Φ. In each round, it adds the user having the maximum $\frac{f(\Phi \cup \{u_i\}) - f(\Phi)}{c_i}$ value into Φ. The algorithm terminates when $f(\Phi) = \frac{m\tau\theta}{\mathcal{T}}$. The computation overhead is dominated by Step 3, which is $O(n^2 m)$.

In addition, there is a small trick in the gDUR algorithm. Note that θ is a constant. It is only related to the approximation factor, and will only be used in the theoretical analysis. Hence, we can simply let $\theta = 1$ in the real implementation of gDUR, since it will not change the comparison results in Steps 2 and 3, and also will not change the final result.

C. Example

To better understand Algorithm 1, we use an example shown in Fig. 3 to illustrate the user recruitment procedure. In the example, $\tau = 1$ day, $\mathcal{T} = 2$ days, $\mathcal{U}$, $\mathcal{S}$, $\mathcal{P}$ and $\mathcal{C}$ are marked in Fig. 3(a). According to Definition 1, $\theta = 15$, and $\frac{m\tau\theta}{\mathcal{T}} = 22.5$. Then, the algorithm is conducted as follows:

- First round: $\Phi = \emptyset$.
- Second round: Since $\Phi = \emptyset$ and $f(\emptyset) = 0 < 22.5$, we first compute $\frac{f(\{u_1\}) - f(\emptyset)}{c_1} = \frac{f(\{u_1\})}{c_1} = 6.75$. Likewise, we have $\frac{f(\{u_2\}) - f(\emptyset)}{c_2} = 4.5$, and $\frac{f(\{u_3\}) - f(\emptyset)}{c_3} = 6.75$. Both u_1 and u_3 can maximize $\frac{f(\Phi \cup \{u_i\}) - f(\Phi)}{c_i}$. Without loss of generality, we add u_1 into Φ.
- Third round: Since $\Phi = \{u_1\}$ this time and $f(\{u_1\}) = 13.5 < 22.5$, we continue the user recruitment procedure by computing $\frac{f(\{u_1, u_2\}) - f(\{u_1\})}{c_2} = 2$. Moreover, we have $\frac{f(\{u_1, u_3\}) - f(\{u_1\})}{c_3} = 4.5$. Since u_3 maximizes $\frac{f(\Phi \cup \{u_i\}) - f(\Phi)}{c_i}$ in this round, we add u_3 into Φ. Now, $\Phi = \{u_1, u_3\}$ and $f(\Phi) = 22.5$. The algorithm terminates and returns the recruited user set $\Phi = \{u_1, u_3\}$.

Fig. 3(b) shows the result $\Phi = \{u_1, u_3\}$ for this user recruitment. It is easy to check that this result is a correct solution. For this example, this solution is even optimal. Moreover, when we let $\theta = 1$, we can get the same result.

D. Correctness

Before the correctness analysis, we first prove an important property of the defined utility function.

***Theorem** 2:* 1) $f(\emptyset) = 0$; 2) $f(\Phi)$ is an increasing function.

Proof: 1) If $\Phi = \emptyset$, then $\rho_j^{\Phi} = 0$ for each $s_j \in \mathcal{S}$, according to Eq. 1. Thus, $f(\Phi = \emptyset) = 0$, according to Definition 1.

2) Without loss of generality, we consider two user sets Φ_1 and Φ_2, where $\Phi_1 \subseteq \Phi_2$. According to Eq. 1, we have $\rho_j^{\Phi_1} \leq \rho_j^{\Phi_2}$ for each $s_j \in \mathcal{S}$. Then, $\min\{\rho_j^{\Phi_1}, \frac{\tau}{\mathcal{T}}\} \leq \min\{\rho_j^{\Phi_2}, \frac{\tau}{\mathcal{T}}\}$. Consequently, we have $f(\Phi_1) = \theta \sum_{j=1}^{m} \min\{\rho_j^{\Phi_1}, \frac{\tau}{\mathcal{T}}\} \leq \theta \sum_{j=1}^{m} \min\{\rho_j^{\Phi_2}, \frac{\tau}{\mathcal{T}}\} = f(\Phi_2)$. Therefore, $f(\Phi)$ is an increasing function. ■

Based on the monotone increasing property of the utility function, we can derive the correctness of the *g*DUR algorithm in the following theorem.

***Theorem** 3:* Algorithm 1 is correct. That is, it will produce a feasible solution of the DUR problem, as long as the problem is solvable. More specifically, 1) Algorithm 1 will terminate for sure; 2) $f(\Phi) = \frac{m\tau\theta}{\mathcal{T}}$ if and only if Φ is a user set who can process each task in $\mathcal{S}$ with an expected completion time no larger than the deadline $\mathcal{T}$.

Proof: 1) For Algorithm 1, in each round of iteration, a user will be added into the recruited user set Φ. In the worst case, all users are recruited after n rounds of iteration. Then, we have $f(\Phi) = f(\mathcal{U}) = \frac{m\tau\theta}{\mathcal{T}}$, and the algorithm will terminate.

2) $\Rightarrow$: According to Definition 1, $f(\Phi) = \frac{m\tau\theta}{\mathcal{T}}$ only when $\min\{\rho_j^{\Phi}, \frac{\tau}{\mathcal{T}}\} = \frac{\tau}{\mathcal{T}}$ for $\forall j \in [1, m]$. This means that $\rho_j^{\Phi} \geq \frac{\tau}{\mathcal{T}}$ for $\forall j \in [1, m]$. Thus, for the arbitrary task s_j in $\mathcal{S}$, the expected completion time satisfies $\frac{\tau}{\rho_j^{\Phi}} \leq \mathcal{T}$.

$\Leftarrow$: If Φ is a set of users who can process each task in $\mathcal{S}$ with an expected completion time no larger than $\mathcal{T}$, we have $\rho_j^{\Phi} \geq \frac{\tau}{\mathcal{T}}$ for $\forall j \in [1, m]$. Consequently, $\min\{\rho_j^{\Phi}, \frac{\tau}{\mathcal{T}}\} = \frac{\tau}{\mathcal{T}}$ for $\forall j \in [1, m]$. Thus, $f(\Phi) = \frac{m\tau\theta}{\mathcal{T}}$ according to Definition 1. ■

Note that $f(\mathcal{U}) = \frac{m\tau\theta}{\mathcal{T}}$. Then, based on Theorem 3, we can equivalently replace the constraint Eq.4 by using $f(\Phi) = f(\mathcal{U})$. That is, we have:

***Corollary** 1:* The DUR problem can be equivalently *reformalized* as:

$$Minimize\{C(\Phi) | f(\Phi) = f(\mathcal{U}), \Phi \subseteq \mathcal{U}\}. \tag{6}$$

E. Performance Analysis

To analyze the approximation ratio of the proposed *g*DUR algorithm, we first show that our DUR problem can be categorized as the MSC/SC problem. Actually, according to [17], a problem can be seen as an MSC/SC problem, if: 1) the problem can be formalized as Eq. 6, i.e., $Minimize\{C(\Phi) | f(\Phi) = f(\mathcal{U}), \Phi \subseteq \mathcal{U}\}$; 2) $f(\Phi)$ is a polymatroid function on $2^{\mathcal{U}}$, i.e., $f(\Phi)$ is submodular and increasing with $f(\Phi) = 0$; 3) $C(\Phi)$ is also a polymatroid function on $2^{\mathcal{U}}$. In the following, we first prove the polymatroid property of $f(\Phi)$.

***Theorem** 4:* $f(\Phi)$ is a submodular function. More specifically, for two arbitrary user sets Φ_1 and Φ_2, $\Phi_1 \subseteq \Phi_2$, and $\forall u_k \in \mathcal{U} \backslash \Phi_2$, the submodular property holds, i.e., $f(\Phi_1 \cup \{u_k\}) - f(\Phi_1) \geq f(\Phi_2 \cup \{u_k\}) - f(\Phi_2)$.

Proof: See Appendix A. ■

Now, we can get the polymatroid property of $f(\Phi)$.

***Theorem** 5:* $f(\Phi)$ is a polymatroid function on $2^{\mathcal{U}}$.

Proof: According to Theorems 2 and 4, $f(\Phi)$ is an increasing, submodular function with $f(\emptyset) = 0$. Hence, we conclude that $f(\Phi)$ is a polymatroid function on $2^{\mathcal{U}}$. ■

Furthermore, we have that $C(\Phi)$ is also polymatroid:

***Theorem** 6:* $C(\Phi)$ is a modular function as well as a polymatroid function on $2^{\mathcal{U}}$.

Proof: $C(\Phi)$ is said to be modular if and only if, for two arbitrary user sets Φ_1 and Φ_2, $C(\Phi)$ satisfies the equation: $C(\Phi_1) + C(\Phi_2) = C(\Phi_1 \cap \Phi_2) + C(\Phi_1 \cup \Phi_2)$. Since $C(\Phi) = \sum_{u_i \in \Phi} c_i$, it is straightforward to verify that the equation holds. Hence, $C(\Phi)$ is a modular function, which implies the submodular property. Moreover, according to $C(\Phi) = \sum_{u_i \in \Phi} c_i$, we have that $C(\Phi)$ is an increasing function with $f(\emptyset) = 0$. Thus, $C(\Phi)$ is also a polymatroid function. ■

According to Corollary 1 and Theorems 5, 6, we can conclude that our DUR problem is an MSC/SC problem. For MSC/SC problems, [17] has proposed a lemma, which can be used to analyze the approximation performance of the proposed *g*DUR algorithm.

***Lemma** 1:* [17] Consider an MSC/SC problem: $Minimize\{C(\Phi) | f(\Phi) = f(\mathcal{U}), \Phi \subseteq \mathcal{U}\}$, in which f is a polymatroid real function on $2^{\mathcal{U}}$, and $f(\mathcal{U}) \geq opt$ where *opt* is the cost of a minimum submodular cover. For a greedy algorithm of this problem, if the selected u_i in each iteration always satisfies $\frac{f(\Phi \cup \{u_i\}) - f(\Phi)}{c_i} \geq 1$, then the greedy solution is a $(1 + \varrho \ln \frac{f(\mathcal{U})}{opt})$-approximation, where if $C(\Phi)$ is modular, then $\varrho = 1$.

Before using Lemma 1 to analyze the approximation ratio of the proposed *g*DUR algorithm, we first show that our utility function $f(\Phi)$ meets the constraints in Lemma 1, as follows:

***Theorem** 7:* Our utility function f satisfies: 1) $f(\mathcal{U}) \geq opt$, where *opt* is the cost of the optimal solution of the DUR problem; 2) in each iteration of Algorithm 1, the selected user u_i satisfies $\frac{f(\Phi \cup \{u_i\}) - f(\Phi)}{c_i} \geq 1$.

Proof: See Appendix B. ■

Based on the above analysis, we can give the approximation ratio of the *g*DUR algorithm by the following theorem.

***Theorem** 8:* The proposed *g*DUR algorithm can achieve a $(1 + \ln \frac{m\tau\theta}{opt\, \mathcal{T}})$-approximation solution, where *opt* is the cost of the optimal solution for the DUR problem.

Proof: According to Lemma 1 and Theorem 7, *g*DUR can achieve a $(1 + \varrho \ln \frac{f(\mathcal{U})}{opt})$-approximation solution, where *opt* is the cost of the optimal solution. According to Theorem 6 and Lemma 1, we have $\varrho = 1$. Thus, the approximation ratio is $1 + \varrho \ln \frac{f(\mathcal{U})}{opt} = 1 + \ln \frac{m\tau\theta}{opt\, \mathcal{T}}$. The theorem holds. ■

V. Extension

In this section, we extend our problem and algorithm to a more practical scenario, in which sensing duration is taken into consideration. We first introduce the extended problem, and then, propose the *d*DUR algorithm to solve this problem, followed by the performance analysis.

A. The Extended Problem

Assume that when a mobile user u_i in $\mathcal{U}$ passes by a PoI to perform the related sensing task s_j $(\in \mathcal{S})$, there is an *average sensing duration* d_{ij} (*minutes* or *hours*) in each sensing cycle, only during which the user u_i can collect the sensing data. Then, the expected sensing duration of user u_i performing task s_j is $d_{ij} p_{ij} \mathcal{T} / \tau$, where $p_{ij} \mathcal{T} / \tau$ is actually the expected number of sensing cycles for user u_i performing task s_j before

Algorithm 2 The dDUR Algorithm

Require: $\mathcal{U}, \mathcal{S}, \mathcal{P}=\{p_{ij}|u_i \in \mathcal{U}, s_j \in \mathcal{S}\}, \mathcal{C}, \tau, \mathcal{T}, \mathcal{D}$

Ensure: Φ

$\Phi = \emptyset$;

while $h(\Phi) < \frac{m\tau\theta}{\mathcal{T}} + \vartheta$ **do**

  Select a user $u_i \in \mathcal{U} \backslash \Phi$ to maximize $\frac{h(\Phi \cup \{u_i\}) - h(\Phi)}{c_i}$;

  $\Phi = \Phi \cup \{u_i\}$;

return Φ;

the deadline. When multiple users in set Φ perform the task s_j, the total expected duration of task s_j is the sum of the expected sensing duration of these users, called *expected sensing time*, denoted by σ_j^{Φ}. That is,

$$\sigma_j^{\Phi} = \sum_{u_i \in \Phi \cap \mathcal{U}_j} d_{ij} p_{ij} \frac{\mathcal{T}}{\tau}. \tag{7}$$

To ensure the quality of sensing data, we let the expected sensing time of task s_j be no less than a threshold $\mathcal{D}$ (≥ 0, unit: *minutes* or *hours*). Then, we have the following extended problem:

$$Minimize: \quad C(\Phi) = \textstyle\sum_{u_i \in \Phi} c_i \tag{8}$$

$$Subject\ to: \quad \Phi \subseteq \mathcal{U} \tag{9}$$

$$\sigma_j^{\Phi} \geq \mathcal{D}, \ 1 \leq j \leq m \tag{10}$$

$$\frac{\tau}{\rho_j^{\Phi}} \leq \mathcal{T}, \ 1 \leq j \leq m \tag{11}$$

B. The dDUR *algorithm*

We adopt the same strategy in the last section to solve the problem. First, we define another utility function for the constraint of sensing duration (i.e., Eq. 10):

***Definition** 2: Utility function* $g(\Phi)$ indicates the utility of sensing duration for the users in set Φ performing all tasks in $\mathcal{S}$, defined as follows:

$$g(\Phi) = \frac{\vartheta}{m\mathcal{D}} \sum_{j=1}^{m} \min\{\sigma_j^{\Phi}, \mathcal{D}\}, \tag{12}$$

where $\vartheta = \theta$ if $\mathcal{D} > 0$; $\vartheta = 0$ and $g(\Phi) = 0$, if $\mathcal{D} = 0$.

Taking both constraints (i.e., Eqs. 10 and 11) into consideration, we combine the two utility functions as follows:

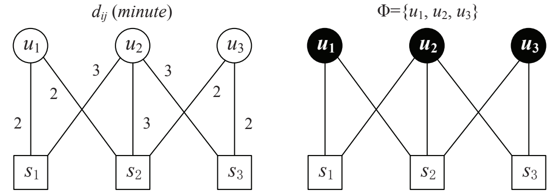
***Definition** 3: Combinational utility function* $h(\Phi)$ is the combination of utility functions $f(\Phi)$ and $g(\Phi)$. That is,

$$h(\Phi) = f(\Phi) + g(\Phi) \tag{13}$$

Based on the combinational utility function $h(\Phi)$, we present the dDUR algorithm, as shown in Algorithm 2. The dDUR algorithm also starts from an empty user set Φ. In each round, it adds the user having the maximum $\frac{h(\Phi \cup \{u_i\}) - h(\Phi)}{c_i}$ value into Φ. The algorithm terminates when $h(\Phi) = \frac{m\tau\theta}{\mathcal{T}} + \vartheta$. The computation overhead is still $O(n^2 m)$.

C. Example

We use the example in Fig. 4 to illustrate the dDUR algorithm. In the example, all d_{ij}'s are marked in Fig. 4(a), and $\mathcal{D} = 3$, with units displayed in minutes. Other parameters



(a) (b)

Fig. 4. Illustration of the dDUR algorithm.

are the same as those in Fig. 3(a). In this example, $\theta = 15$, and $\frac{m\tau\theta}{\mathcal{T}} + \vartheta = 37.5$. Then, the algorithm is conducted as follows:

- First round: $\Phi = \emptyset$ and $h(\emptyset) = 0$.
- Second round: We have $\frac{h(\{u_1\}) - h(\emptyset)}{c_1} = \frac{h(\{u_3\}) - h(\emptyset)}{c_3} = 9.75$, and $\frac{h(\{u_2\}) - h(\emptyset)}{c_2} = 7.5$. Both u_1 and u_3 can maximize $\frac{h(\Phi \cup \{u_i\}) - h(\Phi)}{c_i}$. Then, we add u_1 into Φ.
- Third round: Due to $\Phi = \{u_1\}$ and $h(\{u_1\}) = 19.5 < 37.5$, we continue the user recruitment procedure by computing $\frac{h(\{u_1, u_2\}) - h(\{u_1\})}{c_2} = 4.33$ and $\frac{h(\{u_1, u_3\}) - h(\{u_1\})}{c_3} = 7.33$. Since u_3 maximizes $\frac{h(\Phi \cup \{u_i\}) - h(\Phi)}{c_i}$ in this round, we add u_3 into Φ.
- Fourth round: Now, $\Phi = \{u_1, u_3\}$ and $h(\Phi) = 34.17 < 37.5$. Then, we add the last user u_2 into Φ, to get that $\Phi = \{u_1, u_2, u_3\}$ and $h(\Phi) = 37.5$. The algorithm terminates and returns the recruited user set $\Phi = \{u_1, u_2, u_3\}$.

Fig. 4(b) shows the result $\Phi = \{u_1, u_2, u_3\}$ for this user recruitment procedure. It is easy to check that this result is a correct solution. Moreover, for this example, this solution is the only feasible solution.

D. Performance Analysis

First, we prove the correctness of the dDUR algorithm.

***Theorem** 9:* 1) $h(\Phi)$ is an increasing function with $h(\emptyset) = 0$; 2) $h(\Phi) = \frac{m\tau\theta}{\mathcal{T}} + \vartheta$ if and only if Φ is a feasible solution of the extended DUR problem.

Proof: 1) According to Eqs.7 and 12, $g(\emptyset)$=0, and $g(\Phi)$ is an increasing function. Hence, $h(\Phi)$ is an increasing function with $h(\emptyset) = 0$, according to Theorem 2 and Definition 3.

2) Since $f(\Phi) \leq \frac{m\tau\theta}{\mathcal{T}}$ and $g(\Phi) \leq \vartheta$ according Eqs. 5 and 12, we have $h(\Phi) = \frac{m\tau\theta}{\mathcal{T}} + \vartheta$ if and only if $f(\Phi) = \frac{m\tau\theta}{\mathcal{T}}$ and $g(\Phi) = \vartheta$, which means that Φ satisfies the constraints of Eqs. 11 and 10, respectively. Thus, $h(\Phi) = \frac{m\tau\theta}{\mathcal{T}} + \vartheta$ if and only if Φ is a feasible solution. ■

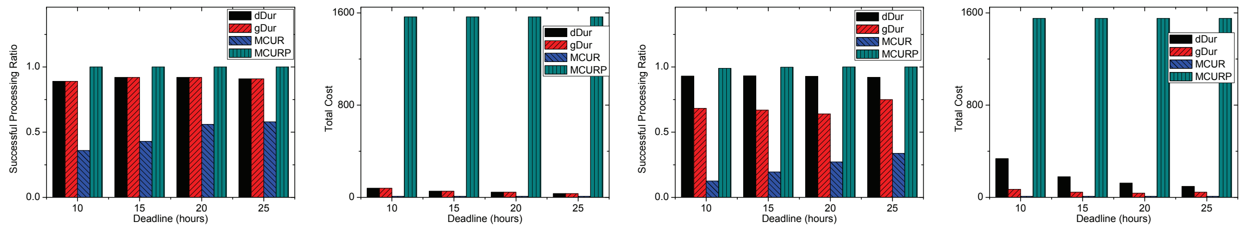
Theorem 9 shows that when the dDUR algorithm terminates, i.e., $h(\Phi) = \frac{m\tau\theta}{\mathcal{T}} + \vartheta$, it will produce a feasible solution. This implies the correctness of dDUR. Moreover, since $h(\mathcal{U}) = \frac{m\tau\theta}{\mathcal{T}} + \vartheta$, we can also re-formalize the extended problem, like Corollary 1.

***Corollary** 2:* The extended DUR problem can be equivalently *re-formalized* as:

$$Minimize\{C(\Phi) | h(\Phi) = h(\mathcal{U}), \Phi \subseteq \mathcal{U}\}. \tag{14}$$

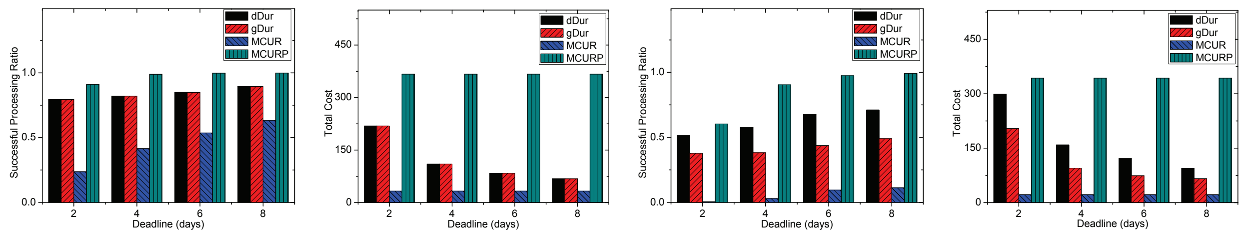
Now, we prove the polymatroid property of $h(\Phi)$. First, we have the following theorem:

***Theorem** 10:* $g(\Phi)$ is a submodular function. More specifically, for two arbitrary user sets Φ_1 and Φ_2, $\Phi_1 \subseteq \Phi_2$, and $\forall u_k \in \mathcal{U} \backslash \Phi_2$, the submodular property holds, i.e., $g(\Phi_1 \cup \{u_k\}) - g(\Phi_1) \geq g(\Phi_2 \cup \{u_k\}) - g(\Phi_2)$.



(a) $\mathcal{D}=0$ minute (b) $\mathcal{D}=0$ minute (c) $\mathcal{D}=4$ minutes (d) $\mathcal{D}=4$ minutes

Fig. 5. Performance comparisons on the real trace: successful processing ratio & total cost vs. deadline ($\mathcal{D}=0, 4$ minutes).



(a) $\mathcal{D}=0$ minute (b) $\mathcal{D}=0$ minute (c) $\mathcal{D}=4$ minutes (d) $\mathcal{D}=4$ minutes

Fig. 6. Performance comparisons on the synthetic trace: successful processing ratio & total cost vs. deadline ($\mathcal{D}=0, 4$ minutes).

Proof: See Appendix C. ∎

Based on Theorem 10, we can get that $h(\Phi)$ is a submodular function, since the linear combination of two submodular functions is still a submodular function. Furthermore, combining this submodular property and Theorem 9, we can straightforwardly have the polymatroid property of $h(\Phi)$:

***Corollary** 3:* $h(\Phi)$ is a polymatroid function on $2^{\mathcal{U}}$.

Based on the above analysis, we can give the approximation ratio of the proposed *d*DUR algorithm as follows:

***Theorem** 11:* The proposed *d*DUR algorithm can achieve a $(1+\ln\frac{m\tau\theta+\mathcal{T}\vartheta}{opt\ \mathcal{T}})$-approximation solution, where *opt* is the cost of the optimal solution for the extended DUR problem.

Proof: According to Theorem 6, Corollaries 2 and 3, the extended DUR problem is an MSC/SC problem. Moreover, since $h(\mathcal{U})=\frac{m\tau\theta}{\mathcal{T}}+\vartheta>\frac{m\tau\theta_1}{\mathcal{T}}=\sum_{i=1}^{n}c_i\geq opt$, and $\frac{h(\Phi\cup\{u_i\})-h(\Phi)}{c_i}=\frac{(f(\Phi\cup\{u_i\})-f(\Phi))+(g(\Phi\cup\{u_i\})-g(\Phi))}{c_i}\geq\frac{f(\Phi\cup\{u_i\})-f(\Phi)}{c_i}\geq 1$, we have that the *d*DUR algorithm can achieve a $(1+\varrho\ln\frac{h(\mathcal{U})}{opt})$-approximation solution, according to Lemma 1. According to Theorem 6 and Lemma 1, we still have $\varrho=1$. Thus, the approximation ratio is $1+\varrho\ln\frac{h(\mathcal{U})}{opt}=1+\ln\frac{m\tau\theta+\mathcal{T}\vartheta}{opt\ \mathcal{T}}$. ∎

VI. Evaluation

We conduct extensive simulations to evaluate the performances of the proposed algorithms. The compared algorithms, the traces that we used, the simulation settings, and the results are presented as follows.

A. Algorithms in Comparison

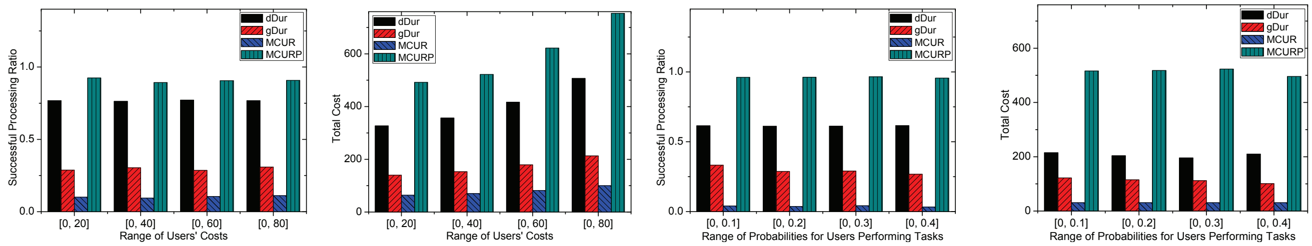
In order to evaluate our algorithms, we implement two other user recruitment algorithms for comparison: MCUR (Minimum Cost User Recruitment) and MCURP (MCUR with Probabilistic mobility). As we discussed in Section II, our problem is different from the existing works, previous user recruitment algorithms cannot be directly applied in our problem. Hence, MCUR and MCURP are designed, mainly based on the idea of the most related algorithms in [7], [10]. Both of them start from an empty user set Φ. In each round, MCUR adds the user u_k, who can maximize the incremental tasks with minimum cost, i.e., $\frac{|\cup_{u_i\in\Phi\cup\{u_k\}}\mathcal{S}_i|-|\cup_{u_i\in\Phi}\mathcal{S}_i|}{c_k}$, into the user set Φ, until all tasks are covered, i.e., $\cup_{u_i\in\Phi}\mathcal{S}_i=\mathcal{S}$. In contrast, MCURP adds the user u_k, who can maximize the effective increments of the probabilities of tasks being processed with minimum cost, i.e., $\frac{\sum_{j=1}^{m}\min\{\sum_{u_i\in\Phi\cup\{u_k\}}p_{ij},1\}-\sum_{j=1}^{m}\min\{\sum_{u_i\in\Phi}p_{ij},1\}}{c_k}$, into the user set Φ, until the additive sum probabilities of all tasks being processed reach 1, i.e., $\min\{\sum_{u_i\in\Phi}p_{ij},1\}=1$ for $\forall s_j\in\mathcal{S}$.

B. The Traces Used and Settings

We adopt the widely-used *Cambridge Haggle Trace Set* [13], which has also been used in [10], [21]. This trace set includes a total of five traces of Bluetooth device connections by people carrying mobile devices (iMotes) over a certain number of days. Among the five traces, we use the trace, generally called *Infocom2006*, in our simulations, since this trace contains some fixed nodes. These can be seen exactly as the PoIs in our model, and it can also provide adequate covers to these fixed nodes. More specifically, the *Infocom2006* trace contains 78 iMotes carried by Infocom 2006 conference participants and 20 fixed nodes situated at various places in the conference hotel, such as conference rooms, the bar, the concierge, and so on.

In our simulations, the set of mobile nodes in the *Infocom2006* trace is mapped to the user set $\mathcal{U}$, and the fixed nodes are mapped to the PoIs as well as the task set $\mathcal{S}$. Moreover, we set the sensing cycle τ as an hour. Then, we extract parts of the trace, and estimate each probability p_{ij} and duration d_{ij} according to the trace. In addition, we generate the cost c_i for each mobile user, which is randomly selected from a cost range $[10, 30]$.

Since the scale of the real trace is very limited, we also randomly generate some synthetic traces in order to make our evaluation more convincing. More specifically, $|\mathcal{U}|$ is selected from $\{100, 200, 300, 400\}$ and $|\mathcal{S}|$ is selected from $\{20, 40, 60, 80\}$. Each sensing duration d_{ij} is randomly selected from a time range $[0.1, 4]$ minutes, and the sensing cycle τ is set as one day.



(a) Ratio vs. Costs of Users (b) Total Cost vs. Costs of Users (c) Ratio vs. Probabilities of Users (d) Total Cost vs. Probabilities of Users

Fig. 7. Performance comparisons on the synthetic traces with the different costs of users, and different probabilities of users performing tasks.

C. Evaluation Metrics, Methods and Results

The major metrics in our simulations include the total cost and the successful processing ratio. The *total cost* is the total cost that the requester needs to pay to all recruited users. The *successful processing ratio* is the ratio of the number of successfully processed crowdsensing jobs and all crowdsensing jobs. Here, a successfully processed crowdsensing job means that the completion time is no larger than the deadline, and the total sensing duration is no less than the given threshold.

To evaluate the performances of *g*DUR, *d*DUR, MCUR, and MCURP, we first conduct two groups of simulations by using the *Infocom2006* trace. In the first group of simulations, we set the deadlines as 10, 15, 20, 25 hours, respectively, and ignore the constraint of sensing duration by setting $\mathcal{D}=0$. In the second group of simulations, we let $\mathcal{D}=4$ minutes. The total costs and successful processing ratios are depicted in Figs. 5(a)-5(d). When $\mathcal{D}=0$, the *d*DUR algorithm will be degraded to *g*DUR, and thus, they achieve the same results. When $\mathcal{D}=4$ minutes, *d*DUR has larger successful processing ratios than *g*DUR, while *g*DUR has smaller total costs than *d*DUR, since *d*DUR takes the sensing duration into consideration, so that it recruits more users than *g*DUR, resulting in larger total costs as well as higher successful processing ratios. Moreover, the results show that *g*DUR and *d*DUR have about 96.7% and 92.4% smaller total costs than MCURP, and also have about 166% and 227% larger successful processing ratios than MCUR, respectively. This is because our algorithms take into consideration both the cost performance and the successful processing ratio metric.

In addition, we also find that the MCUR algorithm has achieved even fewer costs than our algorithms. This is because MCUR does not consider the deadline constraint, so that it recruits many fewer users than our algorithms. Although it produces fewer total costs, it also results in very low successful processing ratios, as shown in Figs. 5(a) and 5(c). Most crowdsensing jobs cannot be completed before the deadlines. On the other hand, we also find that the MCURP algorithm has achieved even larger successful processing ratios than our algorithms in some simulations. This is due to the reason that, in order to ensure the additive sum of probabilities of each task being covered to be larger than 1, MCURP recruits many more users than our algorithms. Although recruiting more users can achieve larger successful processing ratios, it also leads to very large total costs, as shown in Figs. 5(b) and 5(d). Actually, in some simulations, MCURP even recruits all users, resulting in low efficiency. In summary, our algorithms demonstrate much better integrative performances than the compared algorithms.

Second, we also evaluate the performances of the four algorithms by using synthetic traces. In the first group of simulations, we set the deadlines as 2, 4, 6, 8 days, respectively, and let $\mathcal{D}=0$. In the second group of simulations, we set $\mathcal{D}=4$ minutes. In these simulations, each $c_i \in \mathcal{C}$ is randomly selected from $[10, 40]$, and each $p_{ij} \in \mathcal{P}$ is randomly selected from $[0, 0.2]$. The total costs and successful processing ratios are depicted in Figs. 6(a)-6(d). Like the evaluation over the *Infocom2006* trace, the results of both groups' simulations show that *g*DUR and *d*DUR have about 67.6% and 59.0% smaller total costs than MCURP, and also have about 12.2 and 17.0 times larger successful processing ratios than MCUR, respectively, demonstrating much better integrative performances.

Third, we also conduct two additional groups of simulations on the synthetic traces by changing $\mathcal{P}$ and $\mathcal{C}$. In the first group of simulations, we let each $c_i \in \mathcal{C}$ be randomly selected from $[10, 20], [10, 40], [10, 60], [10, 80]$, and $p_{ij} \in \mathcal{P}$ be randomly selected from $[0, 0.1]$. In the second group of simulations, we let each $p_{ij} \in \mathcal{P}$ be randomly selected from $[0, 0.1], [0, 0.2], [0, 0.3], [0, 0.4]$. Each $c_i \in \mathcal{C}$ is randomly selected from $[10, 80]$. Moreover, in these simulations, we set $\mathcal{T}=6$ days and $\mathcal{D}=10$ minutes. The results are shown in Figs. 7(a)-7(d), from which we can derive that *g*DUR and *d*DUR have about 75.0% and 46.2% smaller total costs than MCURP, and also have about 4.33 and 10.9 times larger successful processing ratios than MCUR, respectively. Both *g*DUR and *d*DUR demonstrate much better integrative performances than the two compared algorithms.

VII. Conclusion

We study the DUR problem in the probabilistically collaborative mobile crowdsensing. First, we formalize this problem as a set cover problem with non-linear programming constraints, and prove its NP-hardness. Then, we design a submodular utility function, based on which we propose the greedy approximation algorithm *g*DUR, and derive the corresponding approximation ratio. Moreover, we extend the DUR problem to the case, where sensing duration is taken into consideration, and we propose another approximation algorithm *d*DUR, followed by the analysis of approximation ratio. Extensive simulations based on a real trace and a synthetic trace also verify the performances of the two algorithms.

Acknowledgment

This research was supported in part by the National Natural Science Foundation of China (NSFC) (Grant No. 61572457, 61379132, 61502261, 61303206, 61572342), NSF grants CNS 1449860, CNS 1461932, CNS 1460971, CNS 1439672, CNS 1301774, ECCS 1231461, and the NSF of Jiangsu Province in China (Grant No. BK20131174, BK2009150).

References

[1] G. Cardone, L. Foschini, P. Bellavista, A. Corradi, C. Borcea, M. Talasila, and R. Curtmola. Fostering participaction in smart cities: A geo-social crowdsensing platform. *IEEE Communications Magazine*, 51(6):112–119, 2013.

[2] M. H. Cheung, R. Southwell, F. Hou, and J. Huang. Distributed time-sensitive task selection in mobile crowdsensing. In *ACM MobiHoc*, 2015.

[3] A. Farshad, M. K. Marina, and F. Garcia. Urban wifi characterization via mobile crowdsensing. In *IEEE NOMS*, 2014.

[4] R. K. Ganti, F. Ye, and H. Lei. Mobile crowdsensing: Current state and future challenges. *IEEE Communications Magazine*, 49(11):32–39, 2011.

[5] R. Gao, M. Zhao, T. Ye, F. Ye, Y. Wang, K. Bian, T. Wang, and X. Li. Jigsaw: indoor floor plan reconstruction via mobile crowdsensing. In *ACM MobiCom*, 2014.

[6] S. He, D.-H. Shin, J. Zhang, and J. Chen. Toward optimal allocation of location dependent tasks in crowdsensing. In *IEEE INFOCOM*, 2014.

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

[8] X. Hu, T. H. S. Chu, H. C. B. Chan, and V. C. Leung. Vita: A crowdsensing-oriented mobile cyber-physical system. *IEEE Transactions on Emerging Topics in Computing*, 1(1):148–165, 2013.

[9] L. Jeremie, F. Timur, and C. Vania. Evaluating mobility pattern space routing for dtns. In *IEEE INFOCOM*, 2006.

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

[11] C. Koufogiannakis and N. E. Young. Greedy δ-approximation algorithm for covering with arbitrary constraints and submodular cost. *Algorithmica*, 66(1):113–152, 2013.

[12] E. Koukoumidis, L.-S. Peh, and M. Martonosi. Signalguru: leveraging mobile phones for collaborative traffic signal schedule advisory. In *ACM MobiSys*, 2010.

[13] J. Scott et al. CRAWDAD data set cambridge/ haggle (v. 2009-05-29). Downloaded from http://crawdad. cs.dartmouth. edu/cambridge/haggle, May 2009.

[14] D. Peng, F. Wu, and G. Chen. Pay as how well you do: a quality based incentive mechanism for crowdsensing. In *ACM MobiHoc*, 2015.

[15] X. Sheng, J. Tang, and W. Zhang. Energy-efficient collaborative sensing with mobile phones. In *IEEE INFOCOM*, 2012.

[16] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Performance analysis of mobility-assisted routing. In *ACM MobiHoc*, 2006.

[17] P. Wan, D. Du, P. Pardalos, and W. Wu. Greedy approximations for minimum submodular cover with submodular cost. *Computer Optimization and Applications*, 45(1):463–474, 2010.

[18] Y. Wang, W. Hu, Y. Wu, and G. Cao. Smartphoto: a resource-aware crowdsourcing approach for image sensing with smartphones. In *ACM MobiHoc*, 2014.

[19] Y. Wei, Y. Zhu, H. Zhu, Q. Zhang, and G. Xue. Truthful online double auctions for dynamic mobile crowdsourcing. In *IEEE INFOCOM*, 2015.

[20] M. Xiao, J. Wu, and L. Huang. Home-based zero-knowledge multi-copy routing in mobile social networks. *IEEE Transactions on Parallel and Distributed Systems*, 26(5):1238–1250, 2015.

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

[22] Q. Zhang, Y. Wen, X. Tian, X. Gan, and X. Wang. Incentivize crowd labeling under budget constraint. In *IEEE INFOCOM*, 2015.

[23] X. Zhang, G. Xue, R. Yu, D. Yang, and J. Tang. Truthful incentive mechanisms for crowdsourcing. In *IEEE INFOCOM*, 2015.

[24] Q. Zhao, Y. Zhu, H. Zhu, J. Cao, G. Xue, and B. Li. Fair energy-efficient sensing task allocation in participatory sensing with smartphones. In *IEEE INFOCOM*, 2014.

[25] P. Zhou, Z. Chen, and M. Li. Smart traffic monitoring with participatory sensing. In *ACM SenSys*, 2013.

Appendix

A. Proof of Theorem 4

We first prove that when $|\Phi_2|-|\Phi_1|=1$, $f(\Phi_1\cup\{u_k\})-f(\Phi_1)\geq f(\Phi_2\cup\{u_k\})-f(\Phi_2)$. Then, we extend it to the general case where $|\Phi_2|-|\Phi_1|=w>1$.

First, without loss of generality, we let $\Phi_2\backslash\Phi_1=\{u_h\}$ according to the assumption $\Phi_1\subseteq\Phi_2$ and $|\Phi_2|-|\Phi_1|=1$. To prove the submodular property of $f(\Phi)$, we consider the joint successful processing probability of $\forall s_j\in\mathcal{S}$, which can be divided into the following three cases:

Case 1: u_k cannot process task s_j, i.e., $u_k\notin\mathcal{U}_j$. For this case, $p_{kj}=0$. Then, we have $\rho_j^{\Phi_1\cup\{u_k\}}=\rho_j^{\Phi_1}$ and $\rho_j^{\Phi_2\cup\{u_k\}}=\rho_j^{\Phi_2}$, according to Eq. 1. As a result, $\min\{\rho_j^{\Phi_1\cup\{u_k\}},\frac{\tau}{\mathcal{T}}\}-\min\{\rho_j^{\Phi_1},\frac{\tau}{\mathcal{T}}\}=\min\{\rho_j^{\Phi_2\cup\{u_k\}},\frac{\tau}{\mathcal{T}}\}-\min\{\rho_j^{\Phi_2},\frac{\tau}{\mathcal{T}}\}=0$.

Case 2: u_k can process task s_j, but u_h cannot process this task, i.e., $u_k\in\mathcal{U}_j$ and $u_h\notin\mathcal{U}_j$. For this case, $p_{hj}=0$. According to Eq. 1, $\rho_j^{\Phi_2}=\rho_j^{\Phi_1\cup\{u_h\}}=\rho_j^{\Phi_1}$, and $\rho_j^{\Phi_2\cup\{u_k\}}=\rho_j^{\Phi_1\cup\{u_h,u_k\}}=\rho_j^{\Phi_1\cup\{u_k\}}$. Consequently, we can get $\min\{\rho_j^{\Phi_1\cup\{u_k\}},\frac{\tau}{\mathcal{T}}\}-\min\{\rho_j^{\Phi_1},\frac{\tau}{\mathcal{T}}\}=\min\{\rho_j^{\Phi_2\cup\{u_k\}},\frac{\tau}{\mathcal{T}}\}-\min\{\rho_j^{\Phi_2},\frac{\tau}{\mathcal{T}}\}$.

Case 3: Both u_k and u_h can process task s_j, i.e., $u_k,u_h\in\mathcal{U}_j$. We divide this case into two sub-cases: $p_{kj}\geq p_{hj}$ and $p_{kj}<p_{hj}$. According to Eq. 1, $p_{kj}\geq p_{hj}$ means $\rho_j^{\Phi_1\cup\{u_k\}}\geq\rho_j^{\Phi_1\cup\{u_h\}}$; otherwise, $p_{kj}<p_{hj}$ means $\rho_j^{\Phi_1\cup\{u_k\}}<\rho_j^{\Phi_1\cup\{u_h\}}$.

For the first sub-case, we have $\rho_j^{\Phi_1\cup\{u_k,u_h\}}\geq\rho_j^{\Phi_1\cup\{u_k\}}\geq\rho_j^{\Phi_1\cup\{u_h\}}\geq\rho_j^{\Phi_1}$. Then, we can get:

$$\begin{aligned}&\min\{\rho_j^{\Phi_1\cup\{u_k\}},\frac{\tau}{\mathcal{T}}\}-\min\{\rho_j^{\Phi_1},\frac{\tau}{\mathcal{T}}\}\\&=\begin{cases}p_{kj}(1-\rho_j^{\Phi_1}) & ,\frac{\tau}{\mathcal{T}}\geq\rho_j^{\Phi_1\cup\{u_k,u_h\}};\\ p_{kj}(1-\rho_j^{\Phi_1}) & ,\rho_j^{\Phi_1\cup\{u_k,u_h\}}>\frac{\tau}{\mathcal{T}}\geq\rho_j^{\Phi_1\cup\{u_k\}};\\ \frac{\tau}{\mathcal{T}}-\rho_j^{\Phi_1} & ,\rho_j^{\Phi_1\cup\{u_k\}}>\frac{\tau}{\mathcal{T}}\geq\rho_j^{\Phi_1\cup\{u_h\}};\\ \frac{\tau}{\mathcal{T}}-\rho_j^{\Phi_1} & ,\rho_j^{\Phi_1\cup\{u_h\}}>\frac{\tau}{\mathcal{T}}\geq\rho_j^{\Phi_1};\\ 0 & ,\frac{\tau}{\mathcal{T}}<\rho_j^{\Phi_1}.\end{cases}\end{aligned}\tag{15}$$

$$\begin{aligned}&\min\{\rho_j^{\Phi_2\cup\{u_k\}},\frac{\tau}{\mathcal{T}}\}-\min\{\rho_j^{\Phi_2},\frac{\tau}{\mathcal{T}}\}\\&=\min\{\rho_j^{\Phi_1\cup\{u_k,u_h\}},\frac{\tau}{\mathcal{T}}\}-\min\{\rho_j^{\Phi_1\cup\{u_h\}},\frac{\tau}{\mathcal{T}}\}\\&=\begin{cases}p_{kj}(1-\rho_j^{\Phi_1\cup\{u_h\}}) & ,\frac{\tau}{\mathcal{T}}\geq\rho_j^{\Phi_1\cup\{u_k,u_h\}};\\ \frac{\tau}{\mathcal{T}}-\rho_j^{\Phi_1\cup\{u_h\}} & ,\rho_j^{\Phi_1\cup\{u_k,u_h\}}>\frac{\tau}{\mathcal{T}}\geq\rho_j^{\Phi_1\cup\{u_k\}};\\ \frac{\tau}{\mathcal{T}}-\rho_j^{\Phi_1\cup\{u_h\}} & ,\rho_j^{\Phi_1\cup\{u_k\}}>\frac{\tau}{\mathcal{T}}\geq\rho_j^{\Phi_1\cup\{u_h\}};\\ 0 & ,\rho_j^{\Phi_1\cup\{u_h\}}>\frac{\tau}{\mathcal{T}}\geq\rho_j^{\Phi_1};\\ 0 & ,\frac{\tau}{\mathcal{T}}<\rho_j^{\Phi_1}.\end{cases}\end{aligned}\tag{16}$$

Note that, when $\frac{\tau}{\mathcal{T}}\geq\rho_j^{\Phi_1\cup\{u_k,u_h\}}$, we have $p_{kj}(1-\rho_j^{\Phi_1})-p_{kj}(1-\rho_j^{\Phi_1\cup\{u_h\}})=p_{kj}p_{hj}(1-\rho_j^{\Phi_1})>0$; when $\rho_j^{\Phi_1\cup\{u_k,u_h\}}>\frac{\tau}{\mathcal{T}}\geq\rho_j^{\Phi_1\cup\{u_k\}}$, we can get $p_{kj}(1-\rho_j^{\Phi_1})-(\frac{\tau}{\mathcal{T}}-\rho_j^{\Phi_1\cup\{u_h\}})=p_{kj}p_{hj}(1-\rho_j^{\Phi_1})+\rho_j^{\Phi_1\cup\{u_k,u_h\}}-\frac{\tau}{\mathcal{T}}>0$; and when $\rho_j^{\Phi_1\cup\{u_k\}}>$

$\frac{\tau}{\mathcal{T}} \geq \rho_j^{\Phi_1\cup\{u_h\}}$, we can obtain $\frac{\tau}{\mathcal{T}} - \rho_j^{\Phi_1} - (\frac{\tau}{\mathcal{T}} - \rho_j^{\Phi_1\cup\{u_h\}}) = p_{hj}(1-\rho_j^{\Phi_1}) > 0$, according to Definition 1. Then, comparing Eqs. 15 and 16, we have: $\min\{\rho_j^{\Phi_1\cup\{u_k\}}, \frac{\tau}{\mathcal{T}}\} - \min\{\rho_j^{\Phi_1}, \frac{\tau}{\mathcal{T}}\} \geq \min\{\rho_j^{\Phi_2\cup\{u_k\}}, \frac{\tau}{\mathcal{T}}\} - \min\{\rho_j^{\Phi_2}, \frac{\tau}{\mathcal{T}}\}$.

For the second sub-case, we have $\rho_j^{\Phi_1\cup\{u_k,u_h\}} \geq \rho_j^{\Phi_1\cup\{u_h\}} > \rho_j^{\Phi_1\cup\{u_k\}} \geq \rho_j^{\Phi_1}$. Then, according to Definition 1, we can get:

$$\begin{aligned}
&\min\{\rho_j^{\Phi_1\cup\{u_k\}}, \frac{\tau}{\mathcal{T}}\} - \min\{\rho_j^{\Phi_1}, \frac{\tau}{\mathcal{T}}\} \\
&= \begin{cases} p_{kj}(1-\rho_j^{\Phi_1}) & , \frac{\tau}{\mathcal{T}} \geq \rho_j^{\Phi_1\cup\{u_k,u_h\}}; \\ p_{kj}(1-\rho_j^{\Phi_1}) & , \rho_j^{\Phi_1\cup\{u_k,u_h\}} > \frac{\tau}{\mathcal{T}} \geq \rho_j^{\Phi_1\cup\{u_h\}}; \\ p_{kj}(1-\rho_j^{\Phi_1}) & , \rho_j^{\Phi_1\cup\{u_h\}} > \frac{\tau}{\mathcal{T}} \geq \rho_j^{\Phi_1\cup\{u_k\}}; \\ \frac{\tau}{\mathcal{T}} - \rho_j^{\Phi_1} & , \rho_j^{\Phi_1\cup\{u_k\}} > \frac{\tau}{\mathcal{T}} \geq \rho_j^{\Phi_1}; \\ 0 & , \frac{\tau}{\mathcal{T}} < \rho_j^{\Phi_1}. \end{cases}
\end{aligned} \tag{17}$$

$$\begin{aligned}
&\min\{\rho_j^{\Phi_2\cup\{u_k\}}, \frac{\tau}{\mathcal{T}}\} - \min\{\rho_j^{\Phi_2}, \frac{\tau}{\mathcal{T}}\} \\
&= \min\{\rho_j^{\Phi_1\cup\{u_k,u_h\}}, \frac{\tau}{\mathcal{T}}\} - \min\{\rho_j^{\Phi_1\cup\{u_h\}}, \frac{\tau}{\mathcal{T}}\} \\
&= \begin{cases} p_{kj}(1-\rho_j^{\Phi_1\cup\{u_h\}}) & , \frac{\tau}{\mathcal{T}} \geq \rho_j^{\Phi_1\cup\{u_k,u_h\}}; \\ \frac{\tau}{\mathcal{T}} - \rho_j^{\Phi_1\cup\{u_h\}} & , \rho_j^{\Phi_1\cup\{u_k,u_h\}} > \frac{\tau}{\mathcal{T}} \geq \rho_j^{\Phi_1\cup\{u_h\}}; \\ 0 & , \rho_j^{\Phi_1\cup\{u_h\}} > \frac{\tau}{\mathcal{T}} \geq \rho_j^{\Phi_1\cup\{u_k\}}; \\ 0 & , \rho_j^{\Phi_1\cup\{u_k\}} > \frac{\tau}{\mathcal{T}} \geq \rho_j^{\Phi_1}; \\ 0 & , \frac{\tau}{\mathcal{T}} < \rho_j^{\Phi_1}. \end{cases}
\end{aligned} \tag{18}$$

Note that, when $\frac{\tau}{\mathcal{T}} \geq \rho_j^{\Phi_1\cup\{u_k,u_h\}}$, we still have $p_{kj}(1-\rho_j^{\Phi_1}) - p_{kj}(1-\rho_j^{\Phi_1\cup\{u_h\}}) = p_{kj}p_{hj}(1-\rho_j^{\Phi_1}) > 0$; and when $\rho_j^{\Phi_1\cup\{u_k,u_h\}} > \frac{\tau}{\mathcal{T}} \geq \rho_j^{\Phi_1\cup\{u_h\}}$, we can get $p_{kj}(1-\rho_j^{\Phi_1}) - (\frac{\tau}{\mathcal{T}} - \rho_j^{\Phi_1\cup\{u_h\}}) = p_{kj}p_{hj}(1-\rho_j^{\Phi_1}) + \rho_j^{\Phi_1\cup\{u_k,u_h\}} - \frac{\tau}{\mathcal{T}} > 0$, still according to Definition 1. Then, comparing Eqs. 17 and 18, we also have: $\min\{\rho_j^{\Phi_1\cup\{u_k\}}, \frac{\tau}{\mathcal{T}}\} - \min\{\rho_j^{\Phi_1}, \frac{\tau}{\mathcal{T}}\} \geq \min\{\rho_j^{\Phi_2\cup\{u_k\}}, \frac{\tau}{\mathcal{T}}\} - \min\{\rho_j^{\Phi_2}, \frac{\tau}{\mathcal{T}}\}$.

In summary, $\min\{\rho_j^{\Phi_1\cup\{u_k\}}, \frac{\tau}{\mathcal{T}}\} - \min\{\rho_j^{\Phi_1}, \frac{\tau}{\mathcal{T}}\} \geq \min\{\rho_j^{\Phi_2\cup\{u_k\}}, \frac{\tau}{\mathcal{T}}\} - \min\{\rho_j^{\Phi_2}, \frac{\tau}{\mathcal{T}}\}$ holds for $\forall s_j \in \mathcal{S}$ in all cases. Therefore, $f(\Phi_1\cup\{u_k\}) - f(\Phi_1) \geq f(\Phi_2\cup\{u_k\}) - f(\Phi_2)$.

Now, we consider the case of $|\Phi_2| - |\Phi_1| = w > 1$. Without loss of generality, we assume that $\Phi_2 \backslash \Phi_1 = \{u_h, u_{h+1}, \cdots, u_{h+w-1}\}$. Then, we have $f(\Phi_1 \cup \{u_k\}) - f(\Phi_1) \geq f(\Phi_1 \cup \{u_k, u_h\}) - f(\Phi_1 \cup \{u_h\}) \geq f(\Phi_1 \cup \{u_k, u_h, u_{h+1}\}) - f(\Phi_1 \cup \{u_h, u_{h+1}\}) \geq \cdots \geq f(\Phi_1 \cup \{u_k, u_h, \cdots, u_{h+w-1}\}) - f(\Phi_1\cup\{u_h, \cdots, u_{h+w-1}\}) = f(\Phi_2\cup\{u_k\}) - f(\Phi_2)$.

Therefore, $f(\Phi)$ is a submodular function.

B. Proof of Theorem 7

1) Note that $\mathcal{U}$ is a feasible solution. Hence, we have $f(\mathcal{U}) = \frac{m\tau\theta}{\mathcal{T}}$. Consequently, $f(\mathcal{U}) \geq \frac{m\tau\theta_1}{\mathcal{T}} = \sum_{i=1}^{n} c_i \geq opt$, where opt is the cost of the optimal solution of the DUR problem.

2) Without loss of generality, let u_h be the user recruited in the last round of iteration, and the recruited user set (except u_h) of this round be Φ'. Then, we have $\Phi \subseteq \Phi'$. Moreover, there must be at least a task, whose joint successful processing probability becomes no less than $\frac{\tau}{\mathcal{T}}$ in the last round of iteration; otherwise, the algorithm would have terminated before. Let s_j be such a task. Then, $\rho_j^{\Phi'} < \frac{\tau}{\mathcal{T}}$, while $\rho_j^{\Phi'\cup\{u_h\}} \geq \frac{\tau}{\mathcal{T}}$. Moreover, we have:

$$\begin{aligned}
&\frac{f(\Phi\cup\{u_i\}) - f(\Phi)}{c_i} \\
\geq\ & \frac{f(\Phi\cup\{u_h\}) - f(\Phi)}{c_h} \quad \backslash\backslash u_i \text{ is the optimal selection for } \Phi \\
\geq\ & \frac{f(\Phi'\cup\{u_h\}) - f(\Phi')}{c_h} \quad \backslash\backslash \text{the submodular property of } f(\Phi) \\
\geq\ & \frac{\theta\big(\min\{\rho_j^{\Phi'\cup\{u_h\}}, \frac{\tau}{\mathcal{T}}\} - \min\{\rho_j^{\Phi'}, \frac{\tau}{\mathcal{T}}\}\big)}{c_h} \\
\geq\ & \frac{\theta\big(\frac{\tau}{\mathcal{T}} - \rho_j^{\Phi'}\big)}{c_h} \geq \frac{\theta_2\big(\frac{\tau}{\mathcal{T}} - \rho_j^{\Phi'}\big)}{c_h} \geq 1.
\end{aligned} \tag{19}$$

Therefore, the theorem is correct.

C. Proof of Theorem 10

To prove the submodular property of $g(\Phi)$, we compare $\min\{\sigma_j^{\Phi_1\cup\{u_k\}}, \mathcal{D}\} - \min\{\sigma_j^{\Phi_1}, \mathcal{D}\}$ and $\min\{\sigma_j^{\Phi_2\cup\{u_k\}}, \mathcal{D}\} - \min\{\sigma_j^{\Phi_2}, \mathcal{D}\}$ for $\forall s_j \in \mathcal{S}$.

We first consider the case $u_k \notin \mathcal{U}_j$. According to Eq. 7, we have $\sigma_j^{\Phi_1\cup\{u_k\}} = \sigma_j^{\Phi_1}$ and $\sigma_j^{\Phi_2\cup\{u_k\}} = \sigma_j^{\Phi_2}$. Thus, $\min\{\sigma_j^{\Phi_1\cup\{u_k\}}, \mathcal{D}\} - \min\{\sigma_j^{\Phi_1}, \mathcal{D}\} = \min\{\sigma_j^{\Phi_2\cup\{u_k\}}, \mathcal{D}\} - \min\{\sigma_j^{\Phi_2}, \mathcal{D}\} = 0$.

Then, we consider the case $u_k \in \mathcal{U}_j$. It is simple to verify that σ_j^{Φ} is an increasing function about Φ. Thus, this case can be divided into two sub-cases: $\sigma_j^{\Phi_1} \leq \sigma_j^{\Phi_1\cup\{u_k\}} \leq \sigma_j^{\Phi_2} \leq \sigma_j^{\Phi_2\cup\{u_k\}}$ and $\sigma_j^{\Phi_1} \leq \sigma_j^{\Phi_2} \leq \sigma_j^{\Phi_1\cup\{u_k\}} \leq \sigma_j^{\Phi_2\cup\{u_k\}}$.

For the first sub-case $\sigma_j^{\Phi_1} \leq \sigma_j^{\Phi_1\cup\{u_k\}} \leq \sigma_j^{\Phi_2} \leq \sigma_j^{\Phi_2\cup\{u_k\}}$, we have:

$$\begin{aligned}
&\min\{\sigma_j^{\Phi_1\cup\{u_k\}}, \mathcal{D}\} - \min\{\sigma_j^{\Phi_1}, \mathcal{D}\} \\
&= \begin{cases} d_{kj}p_{kj}\frac{\mathcal{T}}{\tau} & , \mathcal{D} \geq \sigma_j^{\Phi_2\cup\{u_k\}}; \\ d_{kj}p_{kj}\frac{\mathcal{T}}{\tau} & , \sigma_j^{\Phi_2\cup\{u_k\}} > \mathcal{D} \geq \sigma_j^{\Phi_2}; \\ d_{kj}p_{kj}\frac{\mathcal{T}}{\tau} & , \sigma_j^{\Phi_2} > \mathcal{D} \geq \sigma_j^{\Phi_1\cup\{u_k\}}; \\ \mathcal{D} - \sigma_j^{\Phi_1} & , \sigma_j^{\Phi_1\cup\{u_k\}} > \mathcal{D} \geq \sigma_j^{\Phi_1}; \\ 0 & , \mathcal{D} < \sigma_j^{\Phi_1}. \end{cases}
\end{aligned} \tag{20}$$

$$\begin{aligned}
&\min\{\sigma_j^{\Phi_2\cup\{u_k\}}, \mathcal{D}\} - \min\{\sigma_j^{\Phi_2}, \mathcal{D}\} \\
&= \begin{cases} d_{kj}p_{kj}\frac{\mathcal{T}}{\tau} & , \mathcal{D} \geq \sigma_j^{\Phi_2\cup\{u_k\}}; \\ \mathcal{D} - \sigma_j^{\Phi_2} & , \sigma_j^{\Phi_2\cup\{u_k\}} > \mathcal{D} \geq \sigma_j^{\Phi_2}; \\ 0 & , \sigma_j^{\Phi_2} > \mathcal{D} \geq \sigma_j^{\Phi_1\cup\{u_k\}}; \\ 0 & , \sigma_j^{\Phi_1\cup\{u_k\}} > \mathcal{D} \geq \sigma_j^{\Phi_1}; \\ 0 & , \mathcal{D} < \sigma_j^{\Phi_1}. \end{cases}
\end{aligned} \tag{21}$$

Comparing Eqs. 20 and 21, we can get $\min\{\sigma_j^{\Phi_1\cup\{u_k\}}, \mathcal{D}\} - \min\{\sigma_j^{\Phi_1}, \mathcal{D}\} \geq \min\{\sigma_j^{\Phi_2\cup\{u_k\}}, \mathcal{D}\} - \min\{\sigma_j^{\Phi_2}, \mathcal{D}\}$. Likewise, we also have $\min\{\sigma_j^{\Phi_1\cup\{u_k\}}, \mathcal{D}\} - \min\{\sigma_j^{\Phi_1}, \mathcal{D}\} \geq \min\{\sigma_j^{\Phi_2\cup\{u_k\}}, \mathcal{D}\} - \min\{\sigma_j^{\Phi_2}, \mathcal{D}\}$ for the second sub-case. This means that $\min\{\sigma_j^{\Phi_1\cup\{u_k\}}, \mathcal{D}\} - \min\{\sigma_j^{\Phi_1}, \mathcal{D}\} \geq \min\{\sigma_j^{\Phi_2\cup\{u_k\}}, \mathcal{D}\} - \min\{\sigma_j^{\Phi_2}, \mathcal{D}\}$ holds for $\forall s_j \in \mathcal{S}$ in all cases. Therefore, $g(\Phi_1\cup\{u_k\}) - g(\Phi_1) \geq g(\Phi_2\cup\{u_k\}) - g(\Phi_2)$. The theorem is correct.