Dynamic Participant Recruitment of Mobile Crowd Sensing for Heterogeneous Sensing Tasks

Hanshang Li Ting Li Yu Wang
Department of Computer Science
University of North Carolina at Charlotte, Charlotte, NC 28223, USA
{hli39,tli8,yu.wang}@uncc.edu

***Abstract*—With the rapid increasing of smart mobile devices and the advances of sensing technologies, *mobile crowd sensing* (MCS) becomes a new popular sensing paradigm, which enables a variety of large-scale sensing applications. One of the key challenges of large-scale mobile crowd sensing systems is how to effectively select appropriate participants from a huge user pool to perform various sensing tasks while satisfying certain constraints. This becomes more complex when the sensing tasks are dynamic (coming in real time) and heterogeneous (having different temporal and spacial requirements). In this paper, we consider such a dynamic participant recruitment problem with heterogeneous sensing tasks which aims to minimize the sensing cost while maintaining certain level of probabilistic coverage. Both offline and online algorithms are proposed to solve the challenging problem. Extensive simulations over a real-life mobile dataset confirm the efficiency of the proposed algorithms.**

I. Introduction

In recent years, the widespread availability of smart phones equipped with a rich set of built-in sensors has enabled a new paradigm for collecting and sharing sensing data from surrounding environment over a large geographical region: *mobile crowd sensing* (MCS) [1], [2]. Compared with traditional static sensor networks, MCS leverages existing sensing and communication infrastructures without additional costs; provides unprecedented spatio-temporal coverage, especially for observing unpredictable events; integrates human intelligence into the sensing and data processing. These advantages has enabled a broad range of novel MCS applications, such as urban dynamic mining [3], [4], public safety [5] , traffic planning [6]–[9], and environment monitoring [10]–[12].

While large-scale mobile crowd sensing system takes the advantage of huge number participants to enable massive mobile data sensing within urban environments, it also brings many new challenges in the system design. One of the key challenges is participant recruitment problem, how to effectively select appropriate participants from a huge user pool to perform various sensing tasks while satisfying certain constraints. On one hand, more selected participants in MCS can lead to better coverage of sensing tasks over both temporal and spacial domains. On the other hand, the overall sensing cost needs to be minimized. This cost could be the number of selected participants when the cost per participant is fixed. Therefore, careful design of participant recruitment scheme becomes crucial, especially in large-scale MCS system, for the overall performance of crowd sensing and its associated cost. Recently, there are several studies [13]–[18] beginning to address this important issue in MCS. However, some of the methods (such as [17], [18]) only consider the spacial tasks (requiring the coverage of a set of static interested points in spacial domain) and ignore the possibility of temporal requirements of sensing tasks. Some of the other methods (such as [13]–[16]) do consider both temporal and spacial coverage requirements but they assume that the sensing tasks are static (generated before the starting of MCS and no further tasks can come after MCS starts) and with the same length of sensing period. In reality MCS sensing tasks are heterogeneous, they can have different temporal and spacial requirements and various sensing periods. More importantly, the sensing tasks could arrive at any time. Therefore, new dynamic participant recruitment methods are needed for such MSC system with heterogenous sensing tasks.

In this paper, we formulate a new dynamic participant recruitment problem with heterogeneous sensing tasks in a large-scale piggyback MCS system. In a piggyback MCS system [13], [14], the collected sensing data returns to the system by leveraging smartphone usage opportunities to save energy consumption. Therefore, we only focus on the participant recruitment part. We show that finding the minimum participants to achieve certain level of coverage of all tasks is a very challenging problem (actually a NP-hard problem). We then carefully design three greedy algorithms (one offline and two online) to tackle the dynamic participant recruitment problem. Note that since we cannot foreknow when and where a participant will place a phone call during the real crowding sensing period, our proposed methods are based on data driven solution which leverages knowledge obtained via historical call and location traces. We conduct extensive simulations over a real-life mobile dataset (D4D data set [19]) to evaluate the proposed algorithms in different MCS settings. Our results show the proposed methods can achieve stable task coverages while use less number of participants against other simple solutions. We believe that this study is the first on dynamic participant recruitment with heterogeneous sensing tasks in MCS systems.

The rest of this paper is organized as follows. We first introduce our MCS system model and the newly formulated

The work is partially supported by the US National Science Foundation under Grant No. CNS-1319915 and CNS-1343355, and the National Natural Science Foundation of China under Grant No. 61428203.

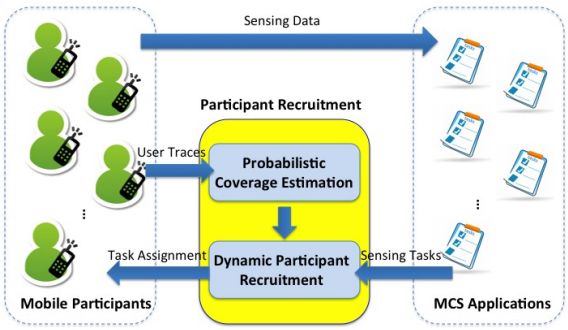
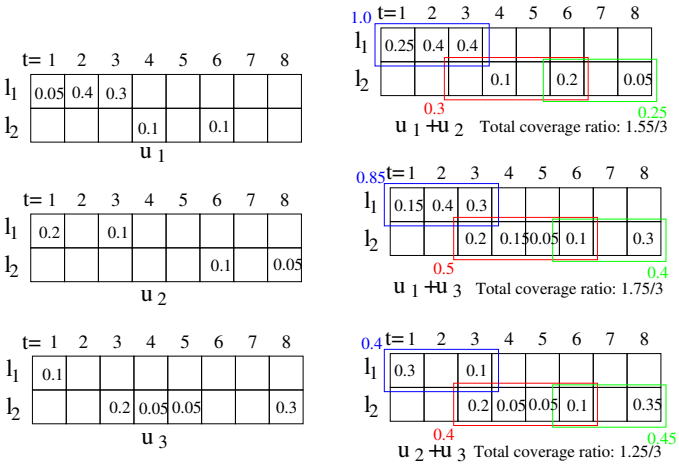


Fig. 1. The framework of dynamic participant recruitment for MCS.



(a) call probability $p(u_i, l_j, t)$ (b) overall coverage, $C(x(i,t))$

Fig. 2. An example with 3 users (u_1, u_2, and u_3), 2 locations (l_1 and l_2), and 3 tasks (with various temporal and spacial coverage marked as blue, red and green rectangles). (a) call probability matrix of each user; (b) coverage ratio of any two users over these three tasks. The colored numbers at corners of tasks are the overall coverage ratios.

participant recruitment problem in Section II. Then we present the detailed design of proposed offline and online algorithms to solve the problem in Section III. Section IV presents our simulation results over a real-life mobile tracing dataset. Section V briefly reviews recent related works on participant selection and task assignment in MCS. Finally, Section VI concludes this paper.

II. MCS System Model and Participant Recruitment Problem

A. System Model

The mobile crowd sensing system includes the following components: a huge number of *mobile participants* who are willing to perform sensing tasks assigned to them, a set of *crowd sensing applications* who are continuously generating *crowd sensing tasks* and looking for sensing data from assigned participants, and the proposed *participant recruitment* component which dynamically decides particular participants for each sensing task. Figure 1 illustrates the overall framework. In this paper, we assume that the task assignment can be sent to each selected participant via cellular service at any time, while the sensing data collected by selected participants will be piggybacked to the mobile crowd sensing system as in [13], [14]. Therefore, we will only focus on the participant selection process.

We assume there are n participant candidates (smartphone users who are registered for participating sensing tasks), denoted as $U = \{u_1, \cdots, u_n\}$, and o sensing locations, denoted by $L = \{l_1, \cdots, l_o\}$. Each user u_i has his own mobility pattern over temporal and spacial domain, which can be described as a predication probability $p(u_i, l_j, t)$, that is the probability of user u_i to place a phone call (or sensing data) at location l_j at time slot t within the whole sensing cycle T (e.g., one or two weeks). For example, Figure 2(a) show three call probability matrices $p(u_i, l_j, t)$ for three users.

A set of m heterogeneous *crowd sensing tasks* $S = \{s_1, \cdots, s_m\}$ generated from the crowd sensing applications. Each of the sensing task s_k can arrive at the system at any time from 1 to T, and we assume that $t_0(s_k)$ is the time slot s_k arrives, and tasks in S are ordered by $t_0(s_k)$. Each task specifies a set of targets for data collection, which includes a location l, a starting time t_s, and an ending time t_e of each target. For each target, the task aims to find a participant to sense the data at location l between time t_s and t_e. In other word, for piggyback crowd sensing if a selected user makes a call at l within the time of $[t_s, t_e]$, we consider that this target is covered or accomplished. To simplify, here we assume that each task only has one single target and the duration of the task (also called life time of the task) is bounded by a fixed value τ (say one day), i.e., $t_e - t_s \leq \tau$. Notice that our proposed methods can be easily extended to handle where sensing tasks with multiple targets. Let $t_s(s_k)$, $t_e(s_k)$, and $l(s_k)$ represent the time and location requirements of task s_k. Figure 2(b) shows three tasks with various lengths in different colors. Then we are ready to formally define the participant recruitment problem.

B. Participant Recruitment Problem

Given the pool of candidates U and the crowd sensing tasks S, the participant recruitment problem aims to minimize total sensing cost while still satisfying certain level of probabilistic coverage of the tasks. The output of the participant recruitment is a set of selected participants to perform the tasks shown by an indicator $x(i,t)$ where $x(i,t) = 1$ if user u_i is selected to participate starting from time t, otherwise $x(i,t) = 0$. Here we assume that when a mobile user is selected to perform sensing task at t, it could cover a fixed time period of τ (e.g., one day). Therefore, we restrict the participant selection of the same user within τ as follows:

$$\sum_{t'=t}^{t+\tau} x(i,t') \leq 1 \textit{ for any } t \in [1,T] \textit{ and } i \in [1,n].$$

Note that a single selected participant can perform the sensing task for multiple tasks and can also be selected multiple times

at different time. Then we can define the cost of sensing task as the summation of all selected participants:

$$\sum_{i\in[1,n]}\sum_{t\in[1,T]} x(i,t).$$

Here the total number of selected participants reflects the total sensing cost (a fix cost per selected participant, such as energy cost of being active for τ). Overall, we would like to minimize the sensing cost.

On the other hand, we also care about the coverage of the sensing tasks. If a selected user u_i makes a call at location $l(s_j)$ within the time of $[t_s(s_j), t_e(s_j)]$, we consider that this task s_j is covered and accomplished. Let $C(i,j,t)$ be the coverage ratio of task s_j by user u_i who is selected starting from t. Then the coverage of task s_j is defined as follows:

$$\min(\sum_{t\in[1,T]}\sum_{i\in[1,n]} C(i,j,t)x(i,t), 1) \textit{ for any } j \in [1,m].$$

Note here if multiple selected users cover the same task, the coverage ratio cannot exceed 1, i.e., fully covered. For example, in Figure 2(b), if u_1 and u_2 are selected, the coverage ratio of the blue task will be 1 even thought the summation of all coverage ratio is larger than 1. Since we cannot foreknow when and where a participant will place a phone call during the crowding sensing period T, we will estimate the $C(i,j,t)$ based on historical call and location traces. We can also define the overall coverage ratio of all task as follows:

$$C(x(i,t)) = \frac{\sum_{j=1}^{m}\min(\sum_{t\in[1,T]}\sum_{i\in[1,n]} C(i,j,t)x(i,t), 1)}{m}.$$

The overall coverage constraint is not a full coverage requirement, instead a probabilistic coverage requirement (i.e., total task coverage is equal to or larger than a predefined coverage threshold γ).

Definition 1: Given the volunteering users U (with their historical call and location traces) and the crowd sensing tasks S, the *Participant Recruitment Problem* is to find participants (i.e., $x(i,t)$) with the objective to

$$\begin{aligned}
&\min_{x} \sum_{i\in[1,n]}\sum_{t\in[1,T]} x(i,t)\\
s.t.\ \ &C(x(i,t)) \geq \gamma\\
&\sum_{t'=t}^{t+\tau} x(i,t') \leq 1 \textit{ for any } t\in[1,T], i\in[1,n]\\
&x(i,t) = 0 \textit{ or } 1 \textit{ for any } t\in[1,T], i\in[1,n].
\end{aligned}$$

Figure 2 shows an example with three users and three tasks. In Figure 2(b), the coverage provided by any two of the three users is provided. It is obvious that choosing u_1 and u_3 leads to best coverage among these three choices. When the numbers of users and tasks are huge, solving this newly defined *participant recruitment problem* (PRP) is a computationally difficult task even when $C(i,j,t)$ is known. We can prove that this problem is NP-hard.

Theorem 1: The *Participant Recruitment Problem* (PRP) is NP-hard.

Proof: This can be obtained from the reduction of the minimum set cover (MSC) problem. Given an instance of MSC, we can construct an instance of PRP as follows. For the set of elements in MSC, we treat them as locations in PRP. Then for each of the subsets in MSC, we create a mobile user who can visit the locations whose corresponding elements are within this subset. Only one sensing task is defined as visiting all locations. Let $T = \tau = 1$ and $\gamma = 1$, we now have a PRP constructed where its optimal solution provide a optimal solution of MSC. Such construction can be done in polynomial time. Since MSC is a well-mown NP-hard problem, PRP is also NP-hard. ■

III. Participant Recruitment Algorithms

In this section, we introduce our proposed participant recruitment algorithms for PRP. Hereafter, we assume that the number of participant candidates are large enough so that if all of them are selected to participant then the sensing tasks can all be fulfilled. Such an assumption is reasonable for large-scale crowd sensing. We first show how we estimate the call probability $p(u_i, l_j, t)$ of a particular user and predict the task coverage ratio $C(i,j,t)$ for any task based on a data driven approach.

A. Estimation of Call Probability and Coverage Ratio

The call probability $p(u_i, l_j, t)$ of a particular user u_i to make a phone call at location l_j and time t is a critical and necessary knowledge for participant recruitment. Since we cannot foreknow when and where a participant will place a phone call during the real crowding sensing period T (e.g., one week), we have to leverage learning from the historical call and location traces. Here, we assume that for each user we have multiple rounds of call traces (e.g., K weeks) in the historical data, and each round of data denoted as D_i, $i = 1, \cdots, K$. Let $X_k(u_i, l_j, t)$ represent whether user u_i made one or more phone call at location l_j and time t in D_i (1 if it made, 0 otherwise). Then we simply estimate the call probability as follow,

$$p(u_i, l_j, t) = \frac{\sum_{k=1}^{K} X_k(u_i, l_j, t)}{K}.$$

Note that we do not consider more complex models where location-transition process is modeled by either Bayesian interferon or Markov model [17] or the call sequence is followed as an inhomogeneous Poisson process [13], [14]. However, such models can be easily integrated into our framework.

In our proposed participant recruitment algorithms, in each round the coverage ratio $C(i,j,t)$ of task s_j by user u_i starting from t is estimated so that we can have a criteria to select individual user. Therefore, we now introduce how we estimate $C(i,j,t)$ from the call probability obtained from historical data. For offline algorithms, the coverage ratio is a summation

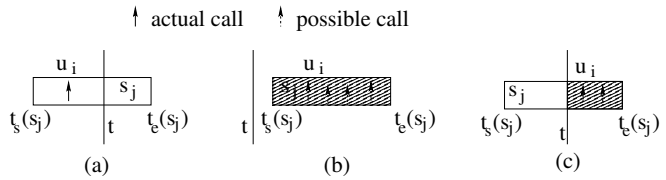


Fig. 3. Three cases of estimation of $C(i,j,t)$ for user u_i and task s_j.

of the call probability of each time unit within the sensing duration τ.

$$C(i,j,t) = \min(\sum_{t'=t}^{t_e(s_j)} p(u_i, l(s_j), t'), 1). \quad (1)$$

In addition, for proposed online algorithms, we also need to estimate the possible coverage $C(i,j,t)$ of the remaining active task s_j from a selected or potential user u_i.

$$C(i,j,t) = \begin{cases} 1 & \text{a call in } [t_s(s_j), t] \\ \min(\sum_{t'=t}^{t_e(s_j)} p(u_i, l(s_j), t'), 1) & \text{no call yet} \end{cases} \quad (2)$$

Note that if a selected user u_i already made a call between $t_s(s_j)$ and current time t as shown in Figure 3(a), task s_j has been covered by u_1 so $C(i,j,t) = 1$. Otherwise, without any call so far, u_i's contribution to task s_j is calculated for the remaining time of this task (as shared areas shown in Figure 3(b) and (c)).

B. Offline Algorithms

Now we are ready to describe our basic offline greedy algorithm. Here we assume that the algorithm knows the whole set of task S for the whole sensing period T. Then in each round we add one participant into the selected pool by greedily selecting the one with largest increasing of total coverage ratio. For example, there are two candidate users u_1 and u_2 (with call probability shown in Figure 4(a) and (b)) and two tasks (with current coverage from previous selected users shown in Figure 4(c)). The offline algorithm will estimate the coverage ratio if select one of the users (as shown in Figure 4(d) and (c)) and pick the one with larger coverage (i.e., u_1 in this example). Algorithm 1 gives the detail of the algorithm. The time complexity of this algorithm is $O(n^2mT^2)$ since at most nT improvements are tested at each round, each improvement involves at most m tasks, and there are at most nT rounds.

Notice that we can also replace Line 4 of the greedy algorithm with other criteria, such as the most active user with maximal calls or even pure random choice. We test those offline methods in the simulation section too.

C. Online Algorithms

In our offline algorithm, the coverage estimation is based on knowledge learned from historical data. However, whether a user make a call at the real sensing period is a random event and the prediction could be wrong. Thus, the actual coverage ratio could be much less than our estimation. One possible enhancement is to add more participants whenever

Algorithm 1 Offline Participant Recruitment Algorithm

Input: participant pool U, task set S, and call probability $p(u_i, l_j, t)$ for each user in U.
Output: $x(i,t)$.

1: $x(i,t) = 0$ for all i and t
2: **while** $C(x(i,t)) < \gamma$ **do**
3: **for all** $u_i \in U$ and $t \in [1,T]$ and $x(i,t) = 0$ **do**
4: Calculate the improvement of $C(x(i,t))$ by adding u_i at time t, i.e., $x(i,t) = 1$ (Here, the coverage ratio is calculated based on Equation (1))
5: **end for**
6: Select the user u_i at time t who leads to the largest coverage improvement, and set $x(i,t) = 1$
7: **end while**
8: **return** $x(i,t)$

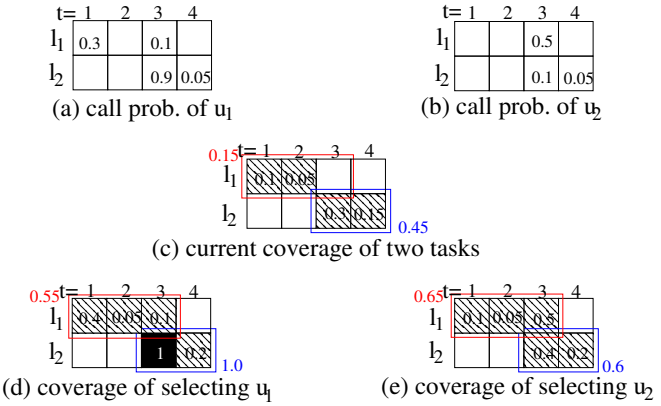


Fig. 4. An example of user selection for two tasks (in red and blue). Shaded cell represents some level of coverage at this cell, while black cell represents full coverage of it. The colored numbers at corners of two tasks are the overall coverage ratios of those tasks.

the estimated sensing coverage is lower than the required ratio. Notice that when time goes by, the coverage ratio of a user to a task may change (as shown in Equation (2), depending on the current time and the remaining duration of the task or whether a call already being made). Therefore, at each time step, our online algorithm will check whether the task has been fulfilled by the current participants. If not, it will greedily selects the new participant who can maximize the coverage improvement (by considering the dynamic of coverage ratio). The details of this algorithm is described in Algorithm 2. The time complexity of this algorithm is $O(n^2mT^3)$. Clearly, this online algorithm needs more selected participants since it adds more participants when the tasking ending time is near and such a task is still not fulfilled.

So far, we always assume that the participant recruitment algorithm knows the whole set of sensing task S beforehand. However, in reality, the tasks are arriving at any moment and the algorithm may not know what kind of tasks will arrive in the future. Therefore, we also provide a real online algorithm to handle such more practice scenario. See Algorithm 3 for detail. Here, at current time slot t, the algorithm will consider

Algorithm 2 Online Participant Recruitment Algorithm with Whole Task Set S

Input: participant pool U, task set S, and call probability $p(u_i, l_j, t)$ for each user in U.
Output: $x(i,t)$.

$x(i,t) = 0$ for all i and t
for $t' = 1$ to T **do**
  while $C(x(i,t)) < \gamma$ **do**
    for all $u_i \in U$ and $t \in [t', T]$ and $x(i,t) = 0$ **do**
      Calculate the improvement of $C(x(i,t))$ by adding u_i at time t, i.e., $x(i,t) = 1$ (Here, the coverage ratio is calculated based on Equation (2))
    end for
    Select the user u_i at time t who leads to the largest coverage improvement, and set $x(i,t) = 1$
  end while
end for
return $x(i,t)$

Algorithm 3 Online Participant Recruitment Algorithm with Current Task Set at Time t

Input: participant pool U, all previous selection $x(i,t')$ for $t' < t$, current task set S_t (including all tasks starting at or before t), and call probability $p(u_i, l_j, t)$ for each user in U.
Output: current selection $x(i,t)$.

Copy all previous selection to $x(i,t)$
while $C(x(i,t)) < \gamma$ based on S_t **do**
  for all $u_i \in U$ and $x(i,t) = 0$ **do**
    Calculate the improvement of $C(x(i,t))$ by adding u_i now at t, i.e., $x(i,t) = 1$ (Here, the coverage ratio is calculated based on Equation (2) and S_t)
  end for
  Select the user u_i who leads to the largest coverage improvement, and set $x(i,t) = 1$
end while
return $x(i,t)$

the current task set S_t which includes both previous started tasks and new tasks arrived at t. If the coverage ratio based on current selection does not reach the requirement, more participants will be selected in the same greedy fashion. Once again the coverage ratio is calculated based on on Equation (2) and only for tasks in S_t. Notice that even there is no new arriving task, the algorithm may still add more participants if the coverage of current tasks is not good enough. The time complexity of this algorithm for time t is $O(nm)$ since at most n users are selected at t and each user at most contributes to m tasks. It seems that this algorithm may not achieve the same level of coverage as Algorithm 2 since the decisions made here are without the knowledge of future incoming tasks. However, our simulation results show the opposite. This is due to that the pure online algorithm has to add enough participants to fulfill the coverage of current task set, which leads to more selected participants and also better coverage.

IV. Simulations

In this section, we conduct extensive simulations over a real-life mobile data (D4D data set [19]) to exam the effectiveness of our proposed greedy algorithms under different participant recruitment scenarios (e.g., online or offline). For the greedy criteria in both offline and online algorithms, we also implement a call activity based and a random one for the comparison with our proposed coverage-based solution. Thus, the following three greedy criteria are tested during the participant recruitment.

- Random: In each round, a random user is selected as the next participant of the MCS.
- Call: In each round, the user with highest call activity is selected as the next participant.
- Coverage: In each round, the user with largest coverage improvement is selected as the next participant.

In total, we implement seven participant recruitment algorithms: *Offline-Coverage* (Algorithm 1), *Offline-Call*, *Offline-Random*, *Online-Coverage* (Algorithm 3), *Online-Call*, *Online-Random*, and *Online-Coverage-TSK* (Algorithm 2) and compare their performance. Notice that *Online-Coverage-TSK* is the online algorithm with full knowledge of the tasks (including future tasks).

In all experiments, we compare each algorithm using the following two measurement metrics.

- Number of selected participants: the number of selected participants[1] generated by the algorithm for the whole task set over the sensing period.
- Number of fulfilled tasks: the number of sensing tasks which are successfully performed by selected participants from the algorithm during the sensing period.

All results reported here are the average from multiple runs over different periods from the D4D data set.

A. D4D Dataset and Simulation Configuration

To simulate the large scale mobile crowd sensing (especially for mobile phone sensing), we use a real life wireless tracing data from the cellular operator Orange for the *Data for Development (D4D) challenge* [20]. The reason we pick the D4D dataset is that it is the only mobile networking tracing dataset available to us which has a large-scale and diverse set of mobile users. The released D4D datasets [19] are based on anonymized Call Detail Records (CDR) of phone calls and SMS exchanges between $50,000$ Orange mobile users in Ivory Coast between December 1, 2011 and April 28, 2012 (150 days and about 20 weeks). Most of the call records are generated between 6:00am to 11:00pm within each single day. We use the dataset of individual trajectories with high spatial

[1]Note that a single user can be selected for multiple sensing periods (each of them lasts τ, e.g. $x(i,t_1) = 1$ and $x(i,t_2) = 1$) and that is counted as multiple participants.

Fig. 5. Locations of cellular towers near Abidjan used as sensing locations in our MCS simulations.

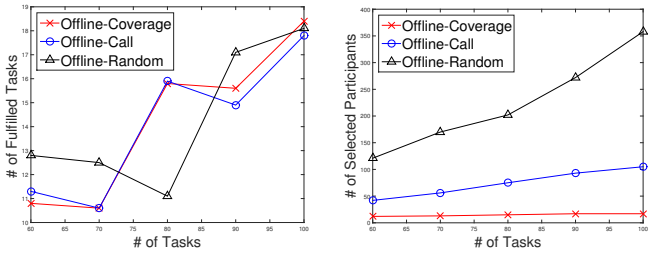
resolution (*SET2* in D4D datasets), which contains 10 groups of the access records of antenna (cellular tower) of each mobile user. Each group of records are collected over a two-week period. The time ranges of these 10 groups of records are sequential and add up equal to the whole duration of D4D data collection period. But unfortunately, in each group of records, the user IDs were renumbered and anonymized, which makes impossible to merge them together. Thus, all of our MCS experiments are preformed within a one week period (i.e., T is one week). We treat one hour as the smallest time unit, $T = 7 \times 24$. We perform simulations over five different weeks. We use the sequences of visited cellular towers of all users within these weeks to generate the call probability of each mobile user and location (i.e., cellular tower). We assume that the mobile users with the same user IDs are same users in all of these weekly call records.

For the D4D dataset, there are already huge number of users and encounters even within a two week period. For example, for the first two-week period, there are $46,254$ active mobile users, $1,097$ cellular towers, and $6,787,594$ encounters between users in total. Therefore, in our simulation, we only choose subsets of users as candidate participants and subsets of cellular towers as locations in MCS. For each sensing task s_i, we randomly pick its location $l(s_i)$, starting time $t_s(s_i)$ and ending time $t_e(s_i)$. For location $l(s_i)$, it is randomly chosen from 20 cellular towers with highest call records. Most of these towers are located in the region of Abidjan, the economic and former official capital of Ivory Coast and the largest city in the nation. Figure 5 shows the locations of these towers on the map of Abidjan. For starting time $t_s(s_i)$, it is randomly chosen from 1 to T. Then ending time $t_e(s_i)$ is randomly chosen from $t_s(s_i)$ to $t_s(s_i)+24$. In other words, the duration of sensing period of a task is limited to one day. For candidate participants, we randomly choose them from the mobile users with the highest number of times of visiting the above towers. All parameters used in the simulations are shown in Table I.

In all simulations, we randomly generate MCS tasks and apply different participant recruitment algorithms to select participants for all tasks. The selected participant will sensing data around the location where he make calls during the assigned time interval (24 hours from the starting time). Based on the real traces, we evaluate how many tasks can be fulfilled with the selected participants. Here, a task is completed if and only if there is at least one call made in the period of the lifetime of the task within the target location from the selected participants. Since the prediction of making a call is based on historical data, it is not possible to guarantee full coverage of all tasks or even the required portion of all tasks.

TABLE I
PARAMETERS USED IN SIMULATIONS

Parameter	Value or Range
Unit of time	1 hour
Task life time $t_e(s_i) - t_s(s_i)$	1 to 24 hours
Number of locations (towers) o	20
Number of tasks m	$60, 70, 80, 90, 100$
Number of candidate participants n	$100, 200, 300, 400, 500$
Length of whole sensing cycle T	one week $= 7 \times 24$ hours
Total simulation period	Dec 5 2011 to Jan 8 2012
Coverage threshold γ	$0.3, 0.4, 0.5, 0.6, 0.7$



(a) (b)

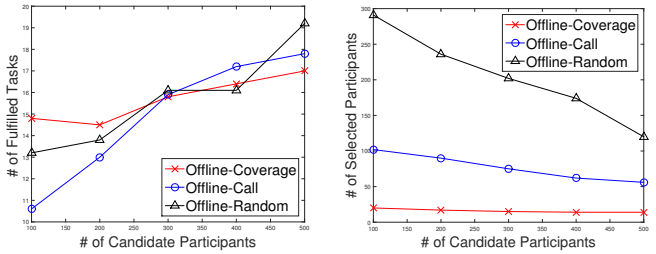
Fig. 6. Results of offline algorithms when $n = 300$ and $m = 60$ to 100.

B. Performance of Offline Algorithms

In the first set of simulations, we compare the performance among different greedy algorithms in the offline setting, in which the full task set S is known and the participant selection is performed offline without further updates. Here, we fix the number of candidate participant at 300, and vary the number of tasks from 60 to 100. Hereafter, the default $\gamma = 0.5$. Figure 6 shows the performance comparison of three offline algorithms.

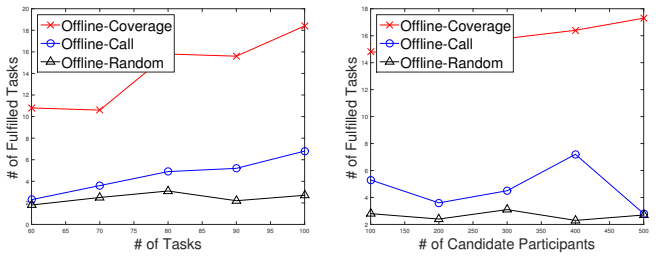
Figure 6(a) shows the number of fulfilled tasks by each algorithm. Clearly, since the selection is based on historical data, the real coverage ratio cannot reach the expected level. However, for *Offline-Coverage* and *Offline-Call* have a clear pattern: the number of fulfilled tasks increases with the number of tasks. This reasonable since the coverage threshold is fixed. *Offline-Random* does not have this pattern. Figure 6(b) shows the number of selected participants. Obviously, the number of selected participants increases with the number of the tasks, i.e., more tasks need more participants. Compared with the three methods, *Offline-Random* uses the largest number of participants while *Offline-Coverage* has the minimum number of participants. The differences among these three methods are significant (*Offline-Random* or *Offline-Call* use 3 or 8 times more participants than *Offline-Coverage* does). This confirms the nice performance of our proposed offline algorithm. Interestedly, the number of participants of *Offline-Coverage* is always less than 20 no matter how many tasks we have. This shows the stability of the proposed method.

In the next set of simulations, we fix the number of task



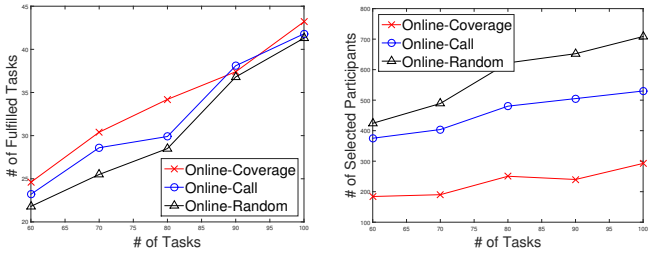
(a) (b)

Fig. 7. Results of offline algorithms when $m = 80$ and $n = 100$ to 500.



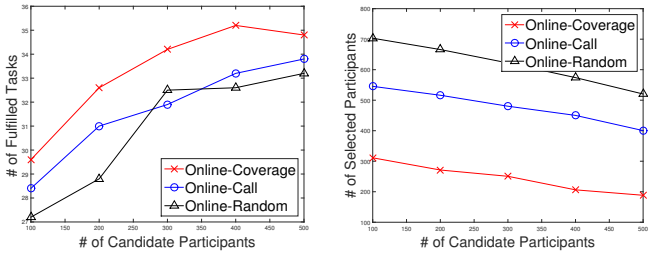
(a) $n = 300$ (b) $m = 80$

Fig. 8. Results of offline algorithms with $n = 300$ or $m = 80$, where *Offline-Random* and *Offline-Call* are forced to select the same number of participants with *Offline-Coverage*.



(a) (b)

Fig. 9. Results of online algorithms when $n = 300$ and $m = 60$ to 100.



(a) (b)

Fig. 10. Results of online algorithms when $m = 80$ and $n = 100$ to 500.

at 80 and test with various number of candidate participant. Figure 7 shows the result. It is clear that the number of fulfilled tasks increases with the number of candidate participants as shown in Figure 7(a), since more candidate participants offers more possible optimized selection. This observation is consist among all methods. The coverage achieved by the three methods are similar. However, in term of the number of selected participants, as shown in Figure 7(b), again *Offline-Coverage* uses much less participants than the other two methods to achieve the same level of coverage. In addition, with more candidate participants, the number of selected participants by all algorithms decreases. This is due to that more candidate participants lead to more space to make smart selections.

Via these two sets of simulations, we found that the task coverages (the numbers of fulfilled tasks) of the three offline methods are similar, but *Offline-Random* and *Offline-Call* select more participants to make up their low efficiency. We also implement another set of simulations to compare with their task coverage with the same number of selected participants. We use the number of selected participants of *Offline-Coverage* as the baseline, and force the other two methods select the same amount of participants. The results are reported in Figure 8. Clearly, now *Offline-Coverage* outperforms the other two methods in term of the number of fulfilled tasks. More precisely, the average numbers of fulfilled tasks of *Offline-Coverage*, *Offline-Call*, and *Offline-Random* are 14.24, 4.56 and 2.46 respectively for simulations with $n = 300$ and various values of m (Figure 8(a)); and 15.78, 4.68 and 2.66 respectively for simulations with $m = 80$ and various values of n (Figure 8(b)).

Overall, the proposed offline algorithm *Offline-Coverage* can achieve better performance than the other two methods. However, since our offline algorithm utilizes estimated call probability to predict the future calls, the achieved coverage ratio is only one third of objective coverage threshold.

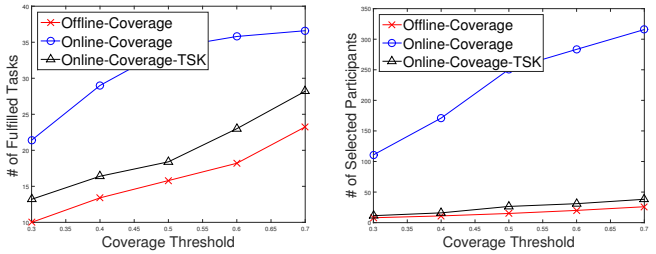
C. Performance of Online Algorithms

To further improve the achieved coverage ratio, we test our online algorithms under the pure online fashion, i.e., the algorithm only knows current sensing task set and have no knowledge of future tasks, but based on the current fulfilled status it can select more participants to achieve the desired coverage ratio. Here, we compare the three type greedy methods. The simulation settings are the same with those in offline tests.

Figure 9 and Figure 10 show the results over two set of simulations where either the number of participants or the number of task is changing. The overall conclusions are similar to those in offline test. First, the number of fulfilled tasks of *Online-Coverage* almost always larger than those of the other two methods. More importantly, it uses much less number of selected participants to achieve such level of coverage. Numbers of selected participants in *Online-Random* and *Online-Call* are almost three times and two times of that in *Online-Coverage*. In addition, compared with offline-methods, online methods output more selected participants in the same task pool. This is due to additional participants are selected at any time to fulfill the unfinished tasks. This improves the coverage ratio but increases the number of selected participants.

D. Offline vs Online Algotihms

Last, we want to further study the difference among offline and online algorithms. Here, we also test the online algorithm with full knowledge of future tasks (i.e. *Online-Coverage-TSK*



(a) (b)

Fig. 11. Results of offline/online algorithms when $m = 80$ and $n = 300$ with various values of γ.

- Algorithm 2 in Section III). Figure 11 shows the results of three algorithms (*Offline-Coverage*, *Online-Coverage*, and *Online-Coverage-TSK*) over the same sets of tasks with different coverage threshold γ (from 0.3 to 0.7). Here, $m = 80$ and $n = 300$. Overall, higher coverage threshold leads to higher number of fulfilled tasks and more selected participants. From Figure 11(a), the number of fulfilled tasks by *Online-Coverage-TSK* is about 50% more than the one by *Offline-Coverage*, while the one by *Online-Coverage* is about two times of that by *Online-Coverage-TSK*. Obviously, *Online-Coverage-TSK* finished the most number of tasks, but it also chooses the largest number of participants as shown in Figure 11(b). *Online-Coverage* selects many times of participants than that in the other two algorithms, while *Online-Coverage-TSK* chooses around 30% more participants than *Offline-Coverage* does.

Summary: From all of the simulation results above, we could draw the following general conclusions. Firstly, the coverage improvement is a better greedy criteria than those based on call activity or purely random selection. Secondly, online algorithms can achieve better coverage than offline algorithms, since they can actively select additional participants. Finally, there is a trade off between the number of selected participants and the coverage level of tasks. It should choose the right algorithm and selection strategy according to particular requirements from the MCS applications.

V. Related Work

With the wide adaptation of mobile crowd sensing applications, task coverage and participant selection in MCS system has drawn many attentions from researchers in recent three years. First, there are several system and experimental studies on either experimental study on MCS coverage [21] or general framework of participant recruitment [22], [23]. For example, Chon *et al.* [21] has preformed a systematic study of the coverage and scaling properties of place-centric urban crowd sensing and shows promising results that MCS can provide relatively high coverage levels especially given area with large size. Then, there are also many theoretical studies on various task assignment and participant selection problems, playing tradeoffs among sensing cost, task coverage, energy efficiency [16], [24], user privacy [18], and incentive [25]–[27]. In most cases, task assignment is equivalent to participant selection. In this section, we only focus on reviewing those who are directly related to our work.

Pournajaf *et al.* [17] also study task assignment in MCS aiming to assign moving participants with uncertain trajectories to static sensing tasks. The optimization goal is to minimize the coverage cost while maximize or maintain certain-level coverage (in term of the number of selected participants per target). The coverage cost is based on the distance between the participant and the task location. An adaptive framework is also proposed to refine the trajectories and perform local task refinement at selected participants. However, no detailed task assignment algorithms are proposed. Then, the same authors also apply similar idea to perform spatial task assignment with cloaked locations to protect the users' privacy in [18], and propose a two-stage task assignment method. However, both these works only consider static spatial tasks (i.e. location-based tasks) which ignore the temporal requirements of sensing tasks and do no allow dynamic tasks.

Zhang *et al.* [15] study offline participant selection in piggyback MCS for probabilistic coverage. They aim to select minimum number of participants to guarantee the selected participants will make enough number of calls at certain percentage of the target locations over a long fixed sensing period. All of their tasks have the same sensing period, and the coverage requirement is different with our model. Similarly, Xiong *et al.* [16] has investigated how to assure the asymptotically full coverage over a 13 tower region with the minimum number of users. Again the task coverage is defined as whether the total number of calls is equal to or more than a threshold at these 13 towers in a fixed time period. They do not consider dynamic heterogenous sensing tasks. Their algorithm predicts the call probability of users to estimate the current coverage, and then allows to assign more participant before the task period ends to enhance the chance of full coverage. Most recently, the same authors [13] further consider a new version of task assignment problem under budget constraint, where the optimization goal is to maximize the number of calls (sensing data) for certain location sets under an overall budget constraint (both base and bonus incentives are given selected participants). However, in all of these works, the sensing tasks are static with fixed time period and no new sensing task can come after the MCS starts.

Overall, we believe that we are the first to study dynamic participant recruitment with heterogeneous sensing tasks. Here, sensing tasks can arrive at any time and may have various temporal/spacial requirements and with various sensing periods. Such problem is much more challenging than those considered in previous studies.

VI. Conclusion

In this paper, we focus on a new dynamic recruitment problem for heterogeneous mobile crowd sensing tasks, with a goal to minimizing the sensing cost while satisfying certain level of coverage. Unlike other existing works, the sensing

tasks in our proposed scenario can have different starting time and life time. Based on the prediction of call probability (the probability of a user making calls at particular time and locations), we propose several offline and online greedy algorithms to dynamically select a subset of participant to perform the tasks. Via extensive simulations conducted with real-life D4D dataset, we confirm the efficiency of our proposed algorithms. We leave further improvements on call prediction as one of our future works.

Acknowledgement: The authors would like to thank Orange and the D4D challenge organizers for providing them the D4D datasets and allowing them to continue working on the datasets after the D4D challenge.

REFERENCES

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

[2] B. Guo, Z. Wang, Z. Yu, Y. Wang, N. Yen, R. Huang, and X. Zhou, "Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm," *ACM Computing Surveys*, to appear.

[3] N. Lathia, V. Pejovic, K. Rachuri, C. Mascolo, M. Musolesi, and P. Rentfrow, "Smartphones for large-scale behavior change interventions," *IEEE Pervasive Computing*, vol. 12, no. 3, pp. 66–73, July 2013.

[4] A. Noulas, S. Scellato, R. Lambiotte, M. Pontil, and C. Mascolo, "A tale of many cities: universal patterns in human urban mobility," *PLOS ONE*, vol. 7, no. 5, p. e37027, 2012.

[5] L. Bengtsson, X. Lu, A. Thorson, R. Garfield, and J. von Schreeb, "Improved response to disasters and outbreaks by tracking population movements with mobile phone network data: A post-earthquake geospatial study in Haiti," *PLoS Med*, vol. 8, no. 8, p. e1001083, 08 2011.

[6] Y. Wang, X. Liu, H. Wei, G. Forman, and Y. Zhu, "Crowdatlas: Self-updating maps for cloud and personal use," in *Proceeding of the 11th Annual ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2013.

[7] P. Zhou, Y. Zheng, and M. Li, "How long to wait?: Predicting bus arrival time with mobile phone based participatory sensing," in *Proceedings of the 10th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2012.

[8] V. Coric and M. Gruteser, "Crowdsensing maps of on-street parking spaces," in *Proceedings of the 2013 IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2013.

[9] S. Nawaz, C. Efstratiou, and C. Mascolo, "Parksense: A smartphone based sensing system for on-street parking," in *Proceedings of the 19th ACM International Conference on Mobile Computing and Networking (MOBICOM 2013)*, 2013.

[10] N. Maisonneuve, M. Stevens, and B. Ochab, "Participatory noise pollution monitoring using mobile phones," *Info. Pol.*, vol. 15, no. 1,2, pp. 51–71, Apr. 2010.

[11] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, "Ear-phone: An end-to-end participatory urban noise mapping system," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2010.

[12] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda, "PEIR, the personal environmental impact report, as a platform for participatory sensing systems research," in *Proceedings of the 7th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2009.

[13] H. Xiong, D. Zhang, L. Wang, J. Gibson, and J. Zhu, "EEMC: Enabling energy-efficient mobile crowdsensing with anonymous participants," *ACM Transactions on Intelligent Systems and Technology (TIST)*, to appear, 2015.

[14] H. Xiong, D. Zhang, G. Chen, L. Wang, and V. Gauthier, "Crowdtasker: Maximizing coverage quality in piggyback crowdsensing under budget constraint," in *IEEE International Conference on Pervasive Computing and Communications(Percom'15)*, 2015.

[15] D. Zhang, H. Xiong, L. Wang, and G. Chen, "Crowdrecruiter: selecting participants for piggyback crowdsensing under probabilistic coverage constraint," in *The 2014 ACM Conference on Ubiquitous Computing*, UbiComp '14, Seattle, WA, USA, 2014, pp. 703–714.

[16] H. Xiong, D. Zhang, L. Wang, and H. Chaouchi, "EMC^3: Energy-efficient data transfer in mobile crowdsensing under full coverage constraint," *IEEE Transactions on Mobile Computing*, to appear, 2015.

[17] L. Pournajaf, L. Xiong, and V. S. Sunderam, "Dynamic data driven crowd sensing task assignment," in *Proceedings of the International Conference on Computational Science, ICCS 2014, Cairns, Queensland, Australia, 10-12 June, 2014*, 2014, pp. 1314–1323.

[18] L. Pournajaf, L. Xiong, V. S. Sunderam, and S. Goryczka, "Spatial task assignment for crowd sensing with cloaked locations," in *IEEE 15th International Conference on Mobile Data Management, MDM 2014, Brisbane, Australia, July 14-18, 2014 - Volume 1*, 2014, pp. 73–82.

[19] V. D. Blondel, M. Esch, C. Chan, F. Clerot, P. Deville, E. Huens, F. Morlot, Z. Smoreda, and C. Ziemlicki, "Data for development: The d4d challenge on mobile phone data," in *arXiv.1210.0137v2*, 2013.

[20] The Data for Development (D4D) Challenge, http://www.d4d.orange.com.

[21] Y. Chon, N. D. Lane, Y. Kim, F. Zhao, and H. Cha, "Understanding the coverage and scalability of place-centric crowdsensing," in *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, 2013, pp. 3–12.

[22] S. Reddy, D. Estrin, and M. Srivastava, "Recruitment framework for participatory sensing data collections," in *Proceedings of the 8th International Conference on Pervasive Computing (Pervasive)*, 2010, pp. 138–155.

[23] G. S. Tuncay, G. Benincasa, and A. Helmy, "Participant recruitment and data collection framework for opportunistic sensing: A comparative analysis," in *Proceedings of the 8th ACM MobiCom Workshop on Challenged Networks (CHANTS)*, 2013, pp. 25–30.

[24] D. Zhao, H. Ma, and L. Liu, "Energy-efficient opportunistic coverage for people-centric urban sensing," *Wireless Networks*, vol. 20, no. 6, pp. 1461–1476, 2014.

[25] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing," in *Proceedings of the 18th Annual ACM International Conference on Mobile Computing and Networking (Mobicom)*, 2012.

[26] Z. Feng, Y. Zhu, Q. Zhang, L. M. Ni, and A. V. Vasilakos, "TRAC: truthful auction for location-aware collaborative sensing in mobile crowdsourcing," in *2014 IEEE Conference on Computer Communications, INFOCOM 2014, Toronto, Canada, April 27 - May 2, 2014*, 2014, pp. 1231–1239.

[27] D. Zhao, X. Li, and H. Ma, "How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint," in *2014 IEEE Conference on Computer Communications, INFOCOM 2014, Toronto, Canada, April 27 - May 2, 2014*, 2014, pp. 1213–1221.