iCrowd: Near-Optimal Task Allocation for Piggyback Crowdsensing

Haoyi Xiong, *Member, IEEE*, Daqing Zhang, Guanling Chen, Leye Wang, Vincent Gauthier, and Laura E. Barnes

Abstract—This paper first defines a novel spatial-temporal coverage metric, k-depth coverage, for mobile crowdsensing (MCS) problems. This metric considers both the fraction of subareas covered by sensor readings and the number of sensor readings collected in each covered subarea. Then *iCrowd*, a generic MCS task allocation framework operating with the energy-efficient Piggyback Crowdsensing task model, is proposed to optimize the MCS task allocation with different incentives and k-depth coverage objectives/constraints. iCrowd first predicts the call and mobility of mobile users based on their historical records, then it selects a set of users in each sensing cycle for sensing task participation, so that the resulting solution achieves two dual optimal MCS data collection goals—i.e., *Goal. 1* near-maximal k-depth coverage without exceeding a given incentive budget or *Goal. 2* near-minimal incentive payment while meeting a predefined k-depth coverage goal. We evaluated iCrowd extensively using a large-scale real-world dataset for these two data collection goals. The results show that: for Goal.1, iCrowd significantly outperformed three baseline approaches by achieving 3-60 percent higher k-depth coverage; for Goal.2, iCrowd required 10.0-73.5 percent less incentives compared to three baselines under the same k-depth coverage constraint.

Index Terms—Mobile crowdsensing (MCS), MCS task allocation, incentives

1 Introduction

With the rapid proliferation of sensor-equipped smartphones, *Mobile Crowdsensing* (MCS) [3] has become an efficient way to sense and collect environment data of urban area in real-time (e.g., air quality, temperature or noise level). Instead of deploying static and expensive sensor network in urban area, MCS leverages the sensors embedded in mobile phones and the mobility of mobile users to sense their surroundings, and utilizes the existing communication infrastructure (e.g., 3G, Wi-Fi etc.) to collect data from mobile phones scattered in the urban area. By collecting sensor readings from mobile users, a "big picture" of the environment in the target area can be obtained using MCS without significant cost.

In this paper we present *iCrowd*—a near-optimal task allocation framework for mobile crowdsensing, which can improve the efficiency of environment data collection with less cost. Here we first discuss the motivations and background of our MCS research, then we formulate a new MCS research problem with a unified set of research assumptions and objectives. We elaborate the technical challenges of the proposed research and finally we summarize our technical contributions.

1.1 Motivations and Backgound

In MCS, there are two main players: *MCS organizer* who is the person or organization coordinating the sensing task, and *MCS participants* who are the mobile users involved in the sensing task. An MCS task usually requires the organizer to recruit participants, to allocate sensing tasks to selected participants, and to collect sensor readings from these participants' mobile devices that well represent the characteristics of the target sensing region [4], often with budget constraints on participant incentives.

Specifically, the MCS organizer needs to specify the target sensing area, which often consists of a set of subareas, and further specify the sensing duration (e.g., 10 days), which is usually divided into equal-length sensing cycles (e.g., each cycle lasts for an hour). Once the settings of subareas and sensing cycles are determined, the MCS application usually needs to collect a number of sensor readings from each subarea of the target region in each sensing cycle. Taking a one-week urban air quality monitoring MCS task as an example, the MCS organizer first divides the whole area into 1 km^2 grid cells, then splits the one-week MCS study time into a sequence of one-hour sensing cycles [5], and further requests at least one MCS participant in each grid to upload the air quality sensor reading in each sensing cycle. In this case, however, the cost of the whole MCS task, including the *energy consumption* caused by the MCS application on each participant's mobile device and the *overall incentives cost* to recruit participants, could be quite high. In order to lower the cost of MCS, the mechanism to reduce the energy consumption and control the overall incentive cost, while ensuring the spatial-temporal

- H. Xiong and L. E. Barnes are with the Department of Systems and Information Engineering, University of Virginia, VA 22904.
E-mail: xhyccc@gmail.com, lb3dp@eservices.virginia.edu.
- D. Zhang is with the SAMOVAR Lab, TELECOM SudParis, Evry, France and Peking University, China. E-mail: daqing.zhang@telecom-sudparis.eu.
- G. Chen is with the Department of Computer Science, University of Massachusetts Lowell, MA 01854. E-mail: glchen@cs.uml.edu.
- L. Wang and V. Gautier are with the SAMOVAR Lab, TELECOM SudParis, Evry, France.
E-mail: {leye.wang, vincent.gauthier}@telecom-sudparis.eu.

Manuscript received 29 May 2015; revised 23 Sept. 2015; accepted 23 Sept. 2015. Date of publication 29 Sept. 2015; date of current version 29 June 2016. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TMC.2015.2483505

coverage of collected environment sensor readings, is thus needed. Next we introduce the background of our research from following thee aspects:

Energy-efficient piggyback crowdsensing (PCS). So far, various solutions have been proposed to reduce energy consumption of individual mobile device, ranging from adapting sensing frequency to inferring part of the data rather than sensing and uploading all data [6], [7]. One of the effective solutions is *Piggyback Crowdsensing*, which reduces energy consumption by leveraging smartphone opportunities to perform sensing tasks and return sensor readings [8], [9]. For example, uploading sensing data in parallel with *a 3G call* can reduce about 75 percent of energy consumption in data transfer compared to the 3G-based solution [9].

Spatial-temporal coverage of MCS tasks. The typical approach for measuring the spatial-temporal coverage is to use the fraction of subareas being covered by at least one sensor reading in each sensing cycle. An MCS application may need to collect sensor readings to achieve either *full spatial-temporal coverage* [10], [11] or *partial spatial-temporal coverage* [12], [13], [14], [15]. Usually, the use of full spatial-temporal coverage is to ensure the collected sensor readings representing each subarea in each sensing cycle, while the use of partial coverage aims to collect data that could represent a certain fraction (e.g., 80 percent) of subareas in each cycle.

Incentives, budget and task allocation. In order to recruit participants for MCS, each selected participant is typically offered a certain amount of money as incentives and thus the MCS organizer needs to prepare a budget equal to the total incentives paid to all participants in each MCS task. Once the spatial-temporal coverage and total budget are determined, the MCS organizer needs to select participants with the goal of either

- *minimizing* the total budget while *ensuring the spatial-temporal coverage*, or
- *maximizing* the spatial-temporal coverage with *a fixed budget*.

In order to achieve either of above goals, given users who are willing to participate the MCS task, an MCS organizer needs to allocate sensing tasks to users, where the organizer first selects participants for MCS [1], and then decides in which sensing cycles each participant should perform the MCS task [14]. Only the participants selected for MCS will be paid with incentives.

With all aforementioned coverage, energy, and incentives issues, we are motivated to study the problem of optimizing MCS tasks, subject to various *spatial-temporal coverage* and *incentive cost* objectives/constraints.

1.2 Assumptions and Objectives

We propose to study a novel MCS task allocation problem for Piggyback Crowdsensing applications, where we first assume that each MCS participant senses and uploads sensor readings leveraging smartphone opportunities (e.g., placing a 3G call) to reduce the MCS energy consumption. We then make following assumptions regarding the spatial-temporal coverage and incentive:

***k** -depth coverage of MCS tasks.* While the existing spatial-temporal coverage metrics usually assume that the environment data (e.g., air quality) of a subarea in a sensing cycle could be represented by a single sensor reading, it is reasonable to believe that the each subarea could be better characterized if we could deduce the environment characteristics using multiple sensor readings collected from the same subarea. However, if we increase the number of sensor readings in a subarea above a certain threshold, the accuracy of the deduced value may not increase anymore [16]. Thus we propose a novel spatial-temporal coverage metrics–i.e., *k-depth coverage*, which could be used as either an objective or a constraint of MCS data collection:

- *K-depth coverage objective:* When using k-depth coverage as an objective of MCS data collection, the MCS application uses the depth k as the *maximum number* of sensor readings desired in each subarea and aims to achieve as higher overall k-depth coverage as possible. Specifically given the depth k, N subareas, and sensor readings obtained in each subarea $s_1, s_{2,N}$, we calculate the k-depth coverage on the $i\text{th}$ subarea as $min\{s_i, k\}$ and the overall k-depth coverage as $\sum_{1\leq i\leq N} min\{s_i, k\}$. For example, if k is set to 3 (i.e., three sensor readings are desired in each subarea during each sensing cycle), the MCS task is said to have the overall k-depth coverage of $0 + 1 + 2 + 3 + 3 = 9$ in a region with five subareas when it receives 0, 1, 2, 3, 4 sensor readings, respectively.
- *K-depth coverage constraint:* When using k-depth coverage as an MCS data collection constraint, the MCS application assumes the depth k as the *minimum number* of sensor readings required in each subarea, and a subarea is said to be covered if at least k sensor readings are collected. However, it might be unnecessary to cover all subareas with k sensor readings, while covering most (e.g., 80 percent) subareas (and each with at least k sensor readings) might be sufficient to characterize the target region. Therefore, we write a k-depth coverage constraint as (k, r), where r refers to the ratio or fraction of subareas each of which is required to receive at least k sensor readings in each cycle. For example, an MCS application can satisfy k-depth coverage constraint $(k = 3, r = 80\%)$, if there exists at least 80 percent subareas covered by at least three sensor readings in each sensing cycle.

Note that the selection of k for an MCS application depends on many factors including the type of sensor readings, the size/shape of the target region, and the local policies of environmental monitoring. Please refer to [17] for the specific k selection.

Base/Bonus incentive models. Instead of providing each participant an equal amount of incentives, it is reasonable to give more incentives to active participants if they are requested to collect sensor readings in more sensing cycles. Thus we adopt a flexible incentive model that consists of the following two components:

- *Base incentive*—a fixed incentive paid to each selected participant (e.g., $50),
- *Bonus incentive*–a varying incentive proportional to the number of sensing cycles assigned (e.g., $1 bonus for participating in one sensing cycle).

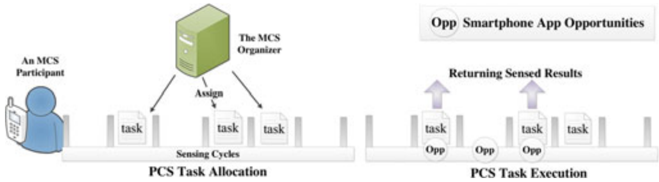


Fig. 1. PCS task allocation and execution.

For example, for the participant shown in Fig. 1 who is involved in three sensing cycles in a PCS task, she would be given $50+$1*3 =$53.

In the context of a PCS task for a target sensing region and the given sensing duration, leveraging k-depth coverage as metrics and using the Base/Bonus Incentive model, we focus on studying following research problems:

Research Objectives–Given a set of mobile users (e.g., 5000 smartphone users), selecting a subset of mobile users as participants, then deciding in which sensing cycle each participant should perform the MCS task, in order to:

- *Goal. 1—maximize the k-depth coverage objective of the PCS system with a fixed amount of incentives,* ***or***
- *Goal. 2—minimize the total amount of incentive payment while ensuring a predefined k-depth coverage constraint.*

As shown in Fig. 1, an MCS participant can sense and upload a sensor reading in a specific sensing cycle if and only if he/she is allocated sensing task in the cycle and encounters a smartphone app opportunity (e.g., places a call) in the cycle. Further, when bonus incentive is set to 0, the task allocation problem becomes a typical user selection problem for either optimization goal,[1] where all sensing cycles of each selected participant should be assigned with sensing tasks to achieve the best task allocation solution.

1.3 Research Examples and Contributions

In order to demonstrate the research challenges of our proposed studies, we use an motivating example as follows: *With the help of a telecom operator, a five-day crowsensing project is launched to monitor the air pollution of a city, updating the air pollution index every hour during daytime with a total budget of 10,000 euros. For the purpose of air quality sensing, the urban area is split into a number of subareas using the cell towers of the telecom operator, where the size of each subarea is less than 1 km^2; Each working day is divided into 10 sensing cycles (08:00–18:00) and each sensing cycle lasts for one hour. In order to characterize the air quality in each subarea, there needs to be at least three sensor readings from each subarea; while collecting more than five sensor readings from each subarea may be redundant. In this way, using the k-depth coverage objective and constraint, two goals for MCS data collection are proposed:*

- *Data Collection Plan.1 (Goal. 1)—Maximize k-depth coverage objectives with $k = 5$ while ensuring the overall incentive payment not exceeding the budget $10,000*; or
- *Data Collection Plan.2 (Goal. 2)—Minimize Overall Incentive Payment while ensuring at least 80 percent subareas being covered by at least three sensor readings, i.e., k-depth coverage constraint with $k = 3$, $r = 80\%$.*

Note that in our work, we mainly focus on the MCS applications for the urban-scale environment monitoring (e.g., air pollution monitoring, noise sensing, or traffic sensing). In this way, we use cell towers as subareas in the rest of this paper. Such setting might not be appropriate for indoor environment monitoring [18] and other MCS applications. Please see also in our appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TMC.2015.2483505, for discussions.

Through the telecom operator, the crowdsensing project successfully makes an agreement with 10,000 smartphone users, who are willing to participate in a five-day air quality sensing trial (i.e., 50 cycles in total) and to install a PCS application [8] on their smartphones. According to the agreement, ***(a)*** *a five-day's call and mobility record of the 10,000 candidates (including the time stamp and cell tower ID for each call) before the trial are made available to the crowdsensing project by the telecom operator;* ***(b)*** *Regarding the base/bonus model, one of following incentive offers might be provided to all participants:*

- **Fixed Incentive Plan—***Each participant is paid equally with a base incentive of $80 (i.e., bonus is set to 0)*, or
- **Varying Incentive Plan—***Each participant is paid with a base incentive of $ 50 and a bonus incentive of $1 for each assigned sensing cycle (i.e., the incentive of each MCS participant is between $51 to $100);*

(c) *the PCS application will sense and upload air quality data when the selected participant places a 3G call at a new subarea in each assigned sensing cycle.* Note that in this paper we assume that each participant returns a sensor reading if he/she places a call in the sensing cycle in which a PCS task is assigned. Besides the call opportunities, piggyback crowdsensing applications can also leverage other smartphone opportunities [8], such as mobile web browsing, to collect and upload sensor readings with low energy cost. Interested readers may find more discussions on this issue in the Appendix, available in the online supplemental material.

Thus with two types of offers and two MCS data collection plans in mind, the PCS tasks might be allocated to participants subject to four possible k-depth coverage and incentive objectives/constraints, which are addressed in Table 1. To optimally allocate tasks subject to any of the four settings, we tackle the following three research challenges: 1) predicting each participants' future call/mobility using the historical call/mobility traces; 2) searching a set of participants for each sensing cycle subject to the overall k-depth coverage/incentive objective/constraint, using the predicted results, and 3) lowering the complexity of participant search while achieving the near-optimal solution, considering the NP-hardness of task allocation problems [1], [2] in nature.

With the above mentioned research objective and examples, the main contributions of this paper are:

1) We formulated the problem of optimal task allocation in piggyback crowdsensing subject to various spatial-temporal coverage and incentive objectives/constraints, with a novel spatial-temporal coverage metric and a flexible incentive model. To the best of our knowledge, this is the first unified framework addressing the task allocation issue in the context of PCS, two dual research objectives are targeted in a

[1] In this paper, we use the term "optimization goal" and "optimal MCS data collection goal" interchangeably.

TABLE 1
Examples of MCS Task Allocation Subject to k-Depth Coverage and Incentive Objectives/Constraints

		Data Collection Plans	
		Plan.1 with Settings: $k = 5$ and $budget = 10,000$	**Plan.2** with Settings: $k = 3$ and $r = 80\%$

Incentive Plans	**Varying**
	Base = $80,
	Bonus = $0
		Selecting $10,000/80 = 125$ MCS participants from all 10,000 mobile users, while maximizing the k-depth coverage objective.	Selecting a minimum number of MCS participants from all 10,000 mobile users while meeting the k-depth coverage constraint.

	Fixed
	Base = $50,
	Bonus = $1
		Selecting $10,000/(50 + 50) = 100$ to $10,000/50 + 1 \approx 196$ MCS participants from all 10,000 mobile users, and for each participant selecting 1 to 50 cycles for task allocation, in order to maximize the k-depth coverage objective under budget constraint.	Selecting 1 to $10,000$ MCS participants, for each participant selecting 1 to 50 sensing cycles for task allocation, in order to minimize overall incentive payment while meeting k-depth coverage constraint

unified manner, leveraging the predicted call/mobility patterns and call opportunities of the participants to sense and upload data in the MCS task.

2) In order to achieve both MCS data collection goals, we proposed a two-phase task allocation framework named iCrowd. It takes a novel approach to search user-cycle combination set, which can achieve either 1) near-maximal k-depth coverage objective under the budget constraint, or 2) near-minimal overall incentive payment under the k-depth coverage constraint. Theoretical analysis shows that the proposed search algorithm can achieve the near-optimality with low computational complexity.

3) We evaluated our proposed algorithms with the real world dataset D4D [19], which contains 4-month call records of 50,000 users from Cote d'Ivoire. We show that the proposed framework performed better than three baseline approaches, using the call records of two separate regions in Abidjan. Specifically, iCrowd achieved 3.0-60 percent higher k-depth coverage on average than the baseline approaches, under the same budget constraint, for Goal. 1 and it also consumed 10.0 - 73.5 percent less overall incentive compared to three baselines under the same k-depth coverage constraint for Goal. 2.

2 Related Work

In order to optimize the overall coverage quality across all sensing cycles with a given incentive budget, two types of task allocations have been studied in the previous work.

Selecting participants for the whole MCS task—Studies in this line of research usually assume each participant is paid with a fixed amount of incentives; then a group of participants are selected to perform the MCS task in all sensing cycles. Reddy et al. first study the research issue of participant selection in participatory sensing, and then propose a coverage-based participant search framework to select a predefined number of participants to maximize the spatial coverage [20], [21]. Singla and Krause propose a novel adaptive participant selection mechanism for maximizing spatial coverage under total incentive constraint in community sensing with respect to privacy [22]. Cardone et al. develop a Mobile Crowdsensing platform, where a simple participant selection mechanism is proposed to maximize the spatial coverage of crowdsensing with a predefined number of participants [23].

Selecting participants for each sensing cycle—Studies in this line of research usually assume each participant is paid with a varied amount of incentives with respect to the number of sensing cycles when the participant performed the MCS task; then for each sensing cycle a subset of participants are selected for the MCS task. In [24], the authors introduce the notion of virtual sensors which intend to collaboratively infer sensing values of each subarea that is not covered by any participant in each sensing cycle, and they propose spatial and temporal coverage quality metrics and leverage the virtual sensor approach in order to reduce the number of participants required in each sensing cycle, while still meeting the coverage quality constraint. Most recently, Hachem et al. [15] proposes a cycle assignment framework for participatory sensing, where the framework predicts mobile users' future locations in next time slot (or sensing cycle) based on their current location and recent trajectory. From the prediction they select a minimal number of mobile users expecting to cover a certain percentage of the target area in the next time slot.

The most relevant work to this paper include [1], [2], [14]. Assuming each participant is given a fixed amount of incentives to complete the whole MCS task, our previous work [1] selects a near-minimal number of participants while ensuring a predefined percentage of subareas being covered by the selected participants, in order to minimize the overall incentive payment under the probabilistic coverage constraint. As the incentive payment for each individual participant is prefixed, we adopted a simple greedy algorithm to approximate the near-optimal solution. On the other hand, assuming each participant is paid a varying amount of incentives according to the number of participated sensing cycles, Zhao et al. [14] select a set of participants from the crowd and determine in which sensing cycle a certain number of participant are needed to perform the MCS task. In order to minimize the overall cost under the data/coverage quality constraint, this work first proposes a novel coverage quality metrics—opportunistic coverage. Given users' historical mobility traces, it then proposes an algorithm to select a small number of participants from volunteers, meeting the predefined opportunistic coverage goal. Finally, an online sampling rate control algorithm, based on real-time location of each participant, decides if each participant should sense and return results in each time slot, to meet the opportunistic coverage quality goals with the minimal number of participants returning sensed results in each time slot. Further, based on a varying incentive payment model, CrowdTasker [2] leverages the historical call/mobility traces of users to select a set of participants for each sensing cycle so as to achieve the near-maximal "coverage quality" under a given budget constraint.

iCrowd is different from above related work in the following aspects. First, compared to our previous work [1], [2] and other task allocation frameworks such as [14], [15], [21], [23], iCrowd is the first unified task allocation solution that can either maximize the k-depth coverage with the given budget or minimize the overall incentive payment under the given k-depth coverage constraint, based on both fixed/varying incentive settings, using the energy-efficient PCS task model. Second, iCrowd selects participants and allocates sensing tasks to participants in specific sensing cycles before the MCS task starts (offline), while [14] selects participants before the MCS task starts but decides if each participant should collect data in a sensing cycle during the MCS task (online). Thus the optimization problems considered in [1], [2], [14], [15], [21], [23] are mathematically different from ours with different objectives and assumptions.

3 iCrowd System Overview

In this section, we formulate the task allocation problem and present the iCrowd framework in detail. Specifically, a generic optimization problem for MCS task allocation is introduced to meet two dual MCS data collection goals; then iCrowd–a unified task allocation framework for achieving both goals—is presented.

3.1 Task Allocation Problem in iCrowd

Given a set of volunteer mobile users, the target region divided by a set of subareas (e.g, cell towers in our study), and the MCS process consisting of a sequence of equal-length sensing cycles (e.g., one cycle per hour), the task allocation problem of iCrowd is to select a number of participants from the volunteer mobile users and to determine in which sensing cycles each selected participant is assigned the PCS task, subject to various optimal MCS data collection goals. With respect to the research objectives introduced in Section 1, we primarily study task allocation problems of the following two goals:

Goal 1. Maximizing k-depth coverage under Budget Constraint – Given a fixed budget for overall incentive payments B, the depth of coverage k, the Base incentive b_a and Bonus incentive b_o, a set of volunteering mobile users $\mathbb{U}$, a target area consisting of a set of cell towers T, the call traces of all users in $\mathbb{U}$ (including the time stamps and associated cell towers of their calls), we denote $\mathbb{S}$ as the set of participants selected from $\mathbb{U}$ (i.e., $\mathbb{S} \subseteq \mathbb{U}$). For each selected participant $\forall u \in \mathbb{S}$, we further denote C_u as a set of cycles assigned to u for PCS task participation (e.g., $C_u = \{0, 2, ...\}$), and $N_{u,i,t}$ as the number of calls made by u at cell tower t in cycle i. This optimization goal is to find $\mathbb{S}$ as a subset of $\mathbb{U}$ and for $\forall u \in \mathbb{S}$ to assign a subset of sensing cycles C_u, with the objective to maximize the overall k-depth coverage with the given budget, i.e.,

$$max. \sum_{0 \leq i < I} \sum_{t \in T} min\left\{ \sum_{u \in \mathbb{S}} min\{N_{u,i,t}, 1\} * \mathbb{A}(C_u, i), k \right\}$$
$$s.t. |\mathbb{S}| * b_a + \sum_{u \in \mathbb{S}} |C_u| * b_o \leq B,$$

where I is the total number of sensing cycles for the PCS task; $\mathbb{A}(C_u, i)$ is a binary function identifying if the participant u is assigned the PCS task in cycle i. Specifically if $i \in C_u$ then $\mathbb{A}(C_u, i) = 1$, if $i \notin C_u$ then $\mathbb{A}(C_u, i) = 0$. Note when the bonus $b_o = 0$, the incentive payment to each participant is fixed (i.e., a Fixed Incentive Setting) at b_a and all her/his cycles should be selected in order to form an optimal solution; thus, the Goal 1. is *equivalent to* selecting a predefined number (i.e., $\lfloor B/b_a \rfloor$) of participants in order to maximize k-depth coverage.

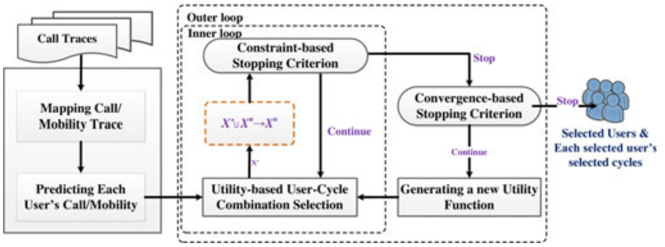

Call Traces
Mapping Call/ Mobility Trace
Predicting Each User's Call/Mobility
Outer loop
Inner loop
Constraint-based Stopping Criterion
X'∪X"→X"
X'
Continue
Stop
Utility-based User-Cycle Combination Selection
Convergence-based Stopping Criterion
Continue
Generating a new Utility Function
Stop
Selected Users & Each selected user's selected cycles

Fig. 2. The icrowd framework.

Goal 2. Minimizing Overall Incentive Payment under k-depth coverage Constraint – Given a predefined k-depth coverage constraint (k, r) where $0 < r \leq 100\%$, this goal is to find $\mathbb{S}$ as a subset of $\mathbb{U}$ and for $\forall u \in \mathbb{S}$ to assign a subset of sensing cycles C_u, with the objective to minimize the total incentive payment while ensuring at least $r * |T|$ subareas being covered by at least k sensed results in each sensing cycle, i.e.,

$$min. |\mathbb{S}| * b_a + \sum_{u \in \mathbb{S}} |C_u| * b_o$$
$$s.t. \sum_{t \in T} max\left\{ \sum_{u \in \mathbb{S}} min\{N_{u,i,t}, 1\} * \mathbb{A}(C_u, i) - k + 1, 0 \right\}$$
$$\geq |T| * r, \text{where } 0 \leq i < I,$$

where I, b_a, b_o, T, and other notations have been defined in *Goal 1*, and $|T| * r$ refers to the minimum number of subareas expected to be covered by at least k sensor readings. Note when the bonus $b_o = 0$ (i.e., a Fixed Incentive Setting), the Goal 2. is *equivalent to* selecting a minimal number of participants while ensuring that the predefined k-depth coverage is achieved.

Next we introduce a generic MCS task allocation framework that can be used to achieve either above two optimization goals, with specific algorithms.

3.2 Overall Design of iCrowd

iCrowd follows a centralized task allocation approach, where a central server collects and stores the volunteering mobile users' historical call traces in the target area, and the server selects participants from all volunteering users ($\mathbb{S} \subseteq \mathbb{U}$) and assigns tasks to each participant in a set of sensing cycles (C_u for each $\forall u \in \mathbb{S}$) before the PCS task execution. Only selected participants are needed to perform sensing tasks, and each selected participant returns sensor readings only in the assigned sensing cycles when a phone call is made. In order to solve the above task allocation problem, iCrowd employs a two-phase solution. In *Phase 1*, it predicts each user's call/mobility in the study period, using the historical call and mobility traces of all users. In *Phase 2*, it incrementally selects participants and assigns sensing tasks to each participant in different sensing cycles based on the *prediction results*, the *estimated k-depth coverage* and *incentive cost*. The framework is shown in Fig. 2 and works as follows.

Phase I: Predicting each user's call/mobility using the historical call/mobility traces – Given the call traces of all volunteering mobile users, this phase computes the *call/mobility profile* of each user-i.e., *probability of each user placing at least one call* *at a particular cell tower in a given sensing cycle*. Specifically, iCrowd computes the profile of each user with the following two steps:

- *Mapping Call/Mobility Traces*—Given the historical call/mobility traces of all users, this step maps each user's historical call/mobility traces onto I sensing cycles and T cell towers. Then it counts $\lambda_{u,i,t}$ the average number of calls placed by each user ($u \in \mathbb{U}$) at each cell tower ($t \in T$) in each sensing cycle ($0 \leq i < I$);
- *Predicting each User's Call/Mobility*—Given $\lambda_{u,i,t}$, this step estimates $P_{i,t}(u)$-the probability of the user ($u \in \mathbb{U}$) placing at least one call at each cell tower ($t \in T$) during each sensing cycle ($0 \leq i < I$).

Phase II: Selecting the participants and determining in which sensing cycles the participants are allocated sensing tasks using the unified Nested-Loop Greedy Search Process for the two dual goals –Given the call/mobility profile of each user, the MCS data collection goal with corresponding settings-i.e., Goal. 1 with budget constraint B and depth k, Goal. 2 with k-depth coverage constraint (k, r), as well as the Base/Bonus Incentive Settings b_a and b_o, we propose a *unified* task allocation process—the *Nested-Loop Greedy Search Process* that can approximate the "real incentive cost" of each participant and search the near-optimal set of user-cycle combinations according to the estimated k-depth coverage and cost. In order to estimate the "real incentive cost", the algorithm first uses the *Inner-loop greedy process* to select a set of user-cycle combinations (namely $\mathbb{X}^1$) using an initial utility (i.e., $Utility_1$) and the corresponding constraint (i.e., budget constraint for Goal 1, and k-depth coverage constraint for Goal 2); with $\mathbb{X}^1$, the *outer-loop convergence process* estimates the incentive cost of each user-cycle combination, then generates a new utility function (i.e., $Utility_2$) according to the estimated incentive cost, and further goes back to the *inner loop* using the new utility function. In this way, the *outer-loop convergence process*, in its each iteration (e.g., the nth outer-loop iteration and $n = 2, 3, 4...$), repeats the process of estimating the incentive cost using the last selected set, generating a new utility function (i.e., $Utility_n$ and $n = 2, 3, 4...$), and searching a new set (i.e., $\mathbb{X}^n$ and $n = 2, 3, 4...$) using the *inner-loop greedy process* and the new utility function (i.e., $Utility_{n+1}$), until the near-optimal combination set is obtained. Specifically, the main components of the Nested-Loop Greedy Search Process works as follows:

- *Inner-loop greedy process* –Given**(a)** the full user set $\mathbb{U}$, **(b)** all sensing cycles $0 \leq i < I$, **(c)** an utility function (notated as $Utility_n$ in the nth outer-loop iteration), and **(d)** the MCS data collection goals with settings i.e., B and k for Goal. 1 and (k, r) for Goal. 2, the algorithm adopts a greedy search process in order to select a set of user-cycle combinations $\mathbb{X}^n$ that intends to maximize the given utility function while satisfying the given constraint. Specifically:
    - ○ The algorithm first selects a subset of user-cycle combinations (namely X') using the *Utility-based User-Cycle Combination Selection Algorithm* (shown in Algorithm. 1), then the algorithm adds X' into the already selected set $\mathbb{X}^n$ (i.e., $X' \cup \mathbb{X}^n \rightarrow \mathbb{X}^n$) for future computation;
    - ○ Further the algorithm decides if it should continue selecting/adding new user-cycle combinations using the corresponding *Constraint-based Inner-Loop Stopping Criterion*: i.e., for Goal. 1 using the *budget-based constraint* shown in Algorithm. 2 to determine if the overall incentive payment exceeds the budget, or for Goal. 2 using the k-depth coverage based constraint shown in Algorithm. 3 to determine if participants/cycles already selected meet the predefined k-depth coverage constraint. Note that $cost(\mathbb{X}^n)$ refers to the overall incentive cost by $\mathbb{X}^n$ in both Goal. 1 and Goal. 2 calculation, $P(\mathbb{X}^n, k, r, i)$ refers to the probability of $\mathbb{X}^n$ achieving the coverage constraint (k, r) of Goal. 2 and P_{thr} is the probability threshold.

Algorithm 1. Utility-based User-Cycle Combination Selection Algorithm

```
Input: $\mathbb{X}^n$–the set of user-cycle combinations already selected
       in the nth outer loop, $\mathbb{U}$—the overall set of users,
       $I$—total number of sensing cycles, and $b_o$—the incentive payment for bonus
Output: $\mathbb{X}'$—a new set of user-cycle combinations selected in
        the current inner loop
begin
    /* initialize */
    $X' \leftarrow \emptyset$;
    /* getting all users in $\mathbb{X}^n$ */
    $\mathbb{S}^n \leftarrow getAllUsers(\mathbb{X}^n)$;
    /* getting all possible user cycle-combinations */
    $\mathbb{C} \leftarrow \{\langle u, i\rangle | \forall u \in \mathbb{U}, 0 \leq i < I\}$;
    if $b_o = 0$ then
        /* when $b_o = 0$ (i.e., Fixed Individual Incentive Setting), select a new user
           (with all cycles) having the maximal utility */
        $u' \leftarrow \mathrm{argmax}_{u \in \mathbb{U} \setminus \mathbb{S}^n} \sum_{0 \leq j < I} Utility(\langle u, j\rangle | \mathbb{X}^n)$;
        $X' \leftarrow \{\langle u', j\rangle | 0 \leq j < I\}$;
    else
        /* when $b_o > 0$ (i.e., Varying Individual Incentive Setting), select a new user-
           cycle combination having the maximal utility */
        $\langle u', i'\rangle \leftarrow \mathrm{argmax}_{u \in \mathbb{C} \setminus \mathbb{X}^n} Utility(\langle u, i\rangle | \mathbb{X}^n)$; $X' \leftarrow \{\langle u', i'\rangle\}$;
    return $\mathbb{X}'$;
```

In this way, the *inner-loop greedy process* continues selecting/adding a new subset of user-cycle combinations and deciding whether new user-cycle combinations should be added using the constraint-based stopping criterion, until the corresponding constraint-based stopping criterion algorithm decides to stop selecting new combinations.

- *Convergence-based outer-loop stopping criterion* –Given the selected set of user-cycle combinations (e.g., $\mathbb{X}^n$ in the nth outer-loop iteration), the algorithm decides whether to return the task allocation results or continue for further computation. When $b_o = 0$ – i.e., the incentive to each participant is fixed, the algorithm stops at the *first* outer-loop iteration and returns $\mathbb{X}^1$ directly as the task allocation result. When $b_o > 0$ – i.e., the individual incentive is dependent on the number of participating cycles, the algorithm needs to decide if to return the task allocation result or continue to obtain $\mathbb{X}^{n+1}$, with respect to the two MCS data collection goals:

- ○ *Goal. 1*—The algorithm first estimates the overall k-depth coverage using $kCov(\mathbb{X}^n)$ *namely the expectation estimation for k-depth coverage*, then the algorithm decides to stop and return $\mathbb{X}^{n-1}$ as task allocation results if $kCov(\mathbb{X}^n) \le kCov(\mathbb{X}^{n-1})$ (i.e., achieving the local maximum of k-depth coverage under budget constraint at solution $\mathbb{X}^{n-1}$).
- ○ *Goal. 2*—The algorithm first estimates the total incentive payment using $cost(\mathbb{X}^n) = b_a * |\mathbb{S}^n| + b_o * |\mathbb{X}^n|$ (where $\forall \langle u, i\rangle \in \mathbb{X}^n, \exists u \in \mathbb{S}^n$), then the algorithm decides to stop and return $\mathbb{X}^{n-1}$ as task allocation results if $cost(\mathbb{X}^{n-1}) \le cost(\mathbb{X}^n)$ (i.e., achieving the local minimum of overall incentive cost under k-depth coverage constraint as the solution $\mathbb{X}^{n-1}$).

- Generating *new Utility Function* and Continuing—If $\mathbb{X}^{n+1}$ does not meet the convergence-based stopping criterion corresponding to the specific optimization goal, the algorithm generates a new utility function namely $Utility_{n+1}(\mathbb{X})$ using $\mathbb{X}^n$, and then continues to the next Outer-loop iteration using the new utility function until the convergence is achieved.

Algorithm 2. Budget-based Stopping Criterion

```
Input: $\mathbb{X}^n$, $B$, $b_o$, $b_a$—incentive payment for base
Output: CONTINUE—continue selecting/adding new user-cycle combinations, STOP–stop and quit to the outer-loop
begin
    /* getting all users in $\mathbb{X}^n$ */
    $\mathbb{S}^n \leftarrow getAllUsers(\mathbb{X}^n)$;
    /* estimating the total incentive payment for already selected set */
    $cost(\mathbb{X}^n) \leftarrow b_a * |\mathbb{S}^n| + b_o * |\mathbb{X}^n|$;
    if ($b_a = 0$ and $cost(\mathbb{X}^n) + b_a \le B$) or ($b_a > 0$ and $cost(\mathbb{X}^n) + b_a + b_o \le B$) then
        CONTINUE;
    else
        STOP;
```

After the Nested-Loop Greedy Search Process terminates, all users $\mathbb{S}$ appeared in $\mathbb{X}^{n-1}$ from *Phase II* (where $\forall\langle u, i\rangle \in \mathbb{X}^{n-1}, \exists u \in \mathbb{S}$) are selected as participants and each selected participant $u \in \mathbb{S}$ is allocated sensing tasks in sensing cycles C_u (where $C_u = \{i | 0 \le i < I \text{and} \exists \langle u, i\rangle \exists \mathbb{X}^{n-1}\}$). The designs of the aforementioned $kCov(\mathbb{X}^n)$, $P(\mathbb{X}^n, k, r, i)$, and $Utility(\langle u, i\rangle | \mathbb{X}^n)$ are discussed in the next section.

4 Core Algorithms and Analysis

In this section, we first introduce the core algorithms including: Call/Mobility Prediction, $kCov(\mathbb{X})$ Calculation, Utility Calculation components used for both Goal. 1 and Goal. 2 optimization and $P(\mathbb{X}, k, r, i)$ Calculation for Goal. 2, then we present the theoretical analysis of our algorithms for achieving the two dual goals.

Algorithm 3. k-depth Coverage-based Stopping Criterion

```
Input: $\mathbb{X}^n$, $b_a$, $b_o$, $(k, r)$—the predefined $k$-depth coverage constraint, $P_{thr}$—threshold, and $B$—the budget for incentive payment
Output: CONTINUE, STOP
begin
    for $0 \le i < I$ do
        /* estimating the $k$-depth coverage probability of already selected set in sensing cycle $i$ */
        if $P(\mathbb{X}^n, k, r, i) < P_{thr}$ then
            CONTINUE;

    STOP;
```

4.1 Call/Mobility Prediction

Assuming the call sequence follows an inhomogeneous Poisson process [25], the probability of a user u to place *at least one phone call* at cell tower $t (t \in T)$ in sensing cycle $i (0 \le i < N)$ can be modeled as:

$$P_{i,t}(u) = 1 - e^{-\lambda_{u,i,t}}, \tag{1}$$

where $\lambda_{u,i,t}$ refers to the Poisson intensity, which is estimated as the average number of calls that u has placed at t in the historical traces corresponding to the sensing cycle i. For example, to estimate $\lambda_{u,i,t}$ for sensing cycle i from 08:00 to 09:00, we will count the average number of calls placed by u at t during the same period 08:00-09:00 in historical traces.

4.2 $kCov(\mathbb{X})$ Calculation

Given a set of user-cycle combinations $\mathbb{X}$, which consists of selected participants $\mathbb{S}$ and each selected participant u's sensing cycles C_u for PCS task participation, the k-depth coverage of $\mathbb{X}$ is

$$\begin{aligned} kCov(\mathbb{X}) = \sum_{0 \le i < I} \sum_{\tau \in T} \sum_{\forall U \subseteq \mathbb{S}} min\{|U|, k\} * \prod_{\forall u \in U} \\ (P_{i,t}(u) * \mathbb{A}(C_u, i)) * \prod_{\forall v \in \mathbb{S} \setminus U} (1 - P_{i,t}(v) * \mathbb{A}(C_v, i)), \end{aligned} \tag{2}$$

where U refers to the set of participants probably returning their sensor readings in cycle i and cell tower t, k refers to the number of sensor readings expected to receive in each subarea/cycle, and the function $\mathbb{A}(C_u, i)$ is defined in Section 3.1. To solve Eq. (2), we implemented a low complexity algorithm for Eq. (2) computation using Probability Generating Function [26].

4.3 Utility Calculation

We now describe two types of utility functions $Utility_1$ and $Utility_n$ ($n \ge 2$). $Utility_1$ is used for the first iteration of the Iterative Greedy Process, and a new utility function $Utility_n$ ($n \ge 2$) is generated for each consecutive iteration.

The $Utility_1$ Calculation – Given the set of incrementally selected user-cycle combinations $\mathbb{X}^1$ in the first iteration of *Iterative Greedy Process* ($\mathbb{X}^1 = \emptyset$ for initialization the greedy search process). The utility for adding a user-cycle combination $\langle v, i\rangle$ combining with $\mathbb{X}^1$ is calculated as:

$$\mathit{Utility}_1(\langle v, j\rangle|\mathbb{X}^1) = \mathit{kCov}(\langle v, j\rangle \cup \mathbb{X}^1) - \mathit{kCov}(\mathbb{X}^1), \tag{3}$$

where $\mathit{kCov}(\mathbb{X}^1)$is the estimated k-depth coverage of $\mathbb{X}^1$, and $\mathit{kCov}(\langle v, j\rangle \cup \mathbb{X}^1)$ is the estimated k-depth coverage of the combined set merging $\langle v, j\rangle$ and $\mathbb{X}^1$. Intuitively $\mathit{Utility}_1$ is the k-depth coverage improvement after adding $\langle v, j\rangle$ into $\mathbb{X}^1$.

The $\mathit{Utility}_n$ Calculation ($n \geq 2$) – During $n\mathrm{th}$ iteration of the Iterative Greedy Process, given the selected set of user-cycle combinations $\mathbb{X}^n$, the algorithm computes the utility for adding each user-cycle combination $\langle v, j\rangle$ to the selected set $\mathbb{X}^n$ as:

$$\mathit{Utility}_n(\langle v, j\rangle|\mathbb{X}^n) = \frac{\mathit{kCov}(\langle v, j\rangle \cup \mathbb{X}^n) - \mathit{kCov}(\mathbb{X}^n)}{\mathit{cost}_n\langle v, j\rangle}, \tag{4}$$

where $\mathit{cost}_n\langle v, j\rangle$ is the *modular incentive cost* [27] of the user-cycle combination $\langle v, j\rangle$. Intuitively $\mathit{Utility}_n$ is the k-depth coverage improvement over the incentive cost of allocating a sensing task to a specific user in a specific sensing cycle. $\mathit{cost}_n\langle v, j\rangle$ is computed as:

$$\begin{aligned}\mathit{cost}_n\langle v, j\rangle &= \mathit{cost}(\mathbb{X}^{n-1}) + \sum_{\langle u,i\rangle \in \{\langle v,j\rangle\}\backslash \mathbb{X}^{n-1}} (b_a + b_o)\\ &- \sum_{\langle u,i\rangle \in \mathbb{X}^{n-1}\backslash \{\langle v,j\rangle\}} [\mathit{cost}(\mathbb{X}^{n-1}) - \mathit{cost}(\mathbb{X}^{n-1}\backslash\{\langle u, i\rangle\})],\end{aligned} \tag{5}$$

where $\mathbb{X}^{n-1}$ is the user-cycle combination set selected in the $n-1\mathrm{th}$ iteration of Iterative Greedy Process. The function $\mathit{cost}(\mathbb{X}) = b_a * |S| + b_o * |\mathbb{X}|$ is the total incentive cost of the user-cycle combination set $\mathbb{X}$, where S is the set of participants that appear in $\mathbb{X}$.

4.4 $P(\mathbb{X}, k, r, i)$ Calculation

Given a set of user-cycle combinations $\mathbb{X}$, the probability of a predefined percentage r of subareas covered by at least k sensed results in cycle i using $\mathbb{X}$ is

$$P(\mathbb{X}, k, r, i) = \sum_{\forall \tau \in T}^{|\tau| \geq T^*} \prod_{t \in \tau} P_{k,i,t}(\mathbb{X}) * \prod_{t' \in T\backslash \tau} (1 - P_{k,i,t}(\mathbb{X})), \tag{6}$$

where $T^* = |T| * r$ refers to the minimum number of subareas expected to be covered by $\mathbb{X}$, $P_{k,i,t}(\mathbb{X})$ refers to the probability of at least k participants returning sensed results in subarea t and sensing cycle i, and is calculated as:

$$\begin{aligned}P_{k,i,t}(\mathbb{X}) = \sum_{\forall U \in \mathbb{S}}^{|U| \geq k} &\prod_{u \in U} (P_{i,t}(u) * \mathbb{A}(C_u, i))*\\ &\prod_{v \in \mathbb{S}\backslash U} (1 - P_{i,t}(v) * \mathbb{A}(C_v, i)),\end{aligned} \tag{7}$$

where $\mathbb{S}$ refers to the set of selected participants in $\mathbb{X}$ and for each selected participant $u \in \mathbb{S}$, C_u refers to the set of allocated sensing cycles for PCS task participation.

4.5 Algorithm Analysis

In this section, we present the theoretical analysis of iCrowd in terms of approximation ratio and computational complexity for optimal MCS data collection Goal. 1 and Goal. 2, respectively.

Performance for Goal. 1 – According to the theory of submodular function maximization under the submodular kapsack constraint [27], iCrowd can guarantee a *Near-Optimal* solution with $(\alpha, 1 - e^{-1})$-approximation bound ($\alpha \approx \frac{b_a+bo}{b_a}$) when maximizing $\mathit{kCov}(\mathbb{X})$ with the given budget. For example, given the Base/Bonus incentive settings $b_a = \$50$ and $b_o = \$1$, supposing with \$10,000 budget the optimal solution obtained by the brute-force enumeration algorithm achieves the total coverage quality of 1,000 in expectation, then iCrowd with $\$10000 * \frac{50+1}{50} = \$10200$ budget can achieve at least a coverage quality of 630.

Performance for Goal. 2 – Considering the duality of the submodular maximization/minimization problems [27] between Goal. 1 and Goal. 2, we could easily conclude that iCrowd could achieve near-optimality in minimizing the overall incentive payment under k-depth coverage constraint, using our conclusion made for Goal. 1. For detailed discussion on the duality between the optimization problems of Goal. 1 and Goal. 2, please refer to [27].

For the proofs of submodularity, computational complexity, and approximation bound, please see the Appendix, available in the online supplemental material.

5 Evaluation

In this section, we show the evaluation result of iCrowd for the two dual MCS data collection goals. Specifically, we first introduce the datasets used in the experiments of both goals, and then present the evaluation results of iCrowd for Goal. 1 and Goal. 2, respectively, to compare performances against baselines.

5.1 Dataset and Experiment Setups

The dataset we used in evaluation is the D4D dataset [19], which contains 50,000 users' phone call traces (each call records includes user ID, call time, and cell tower) from Cote d'Ivoire. All these users are re-selected randomly every two weeks with anonymized user IDs. Thus in this study, we design experiments based on such two-week periods. The call traces in the first week were used for participant selection, and we simulated the spatial-temporal coverage of selected participants using call traces in the second week. Specifically, we extract the call traces of two connected regions in four two-week periods and build the following three datasets for our evaluation:

- *BUSINESS* – a commercial center of the city where 86 cell towers having been installed and around 7,945-8,799 mobile phone users placing phone calls in any two-week period.
- *RESIDENTIAL* – a residential area where 45 cell towers having been installed and around 6,034-6,627 mobile phone users placing phone calls in any two-week period.
- *MERGED* – combined area of both BUSINESS and RESIDENTIAL regions where 131 cell towers having been installed and around 11,363-12,049 unique mobile phone users placing phone calls in any two-week slot.

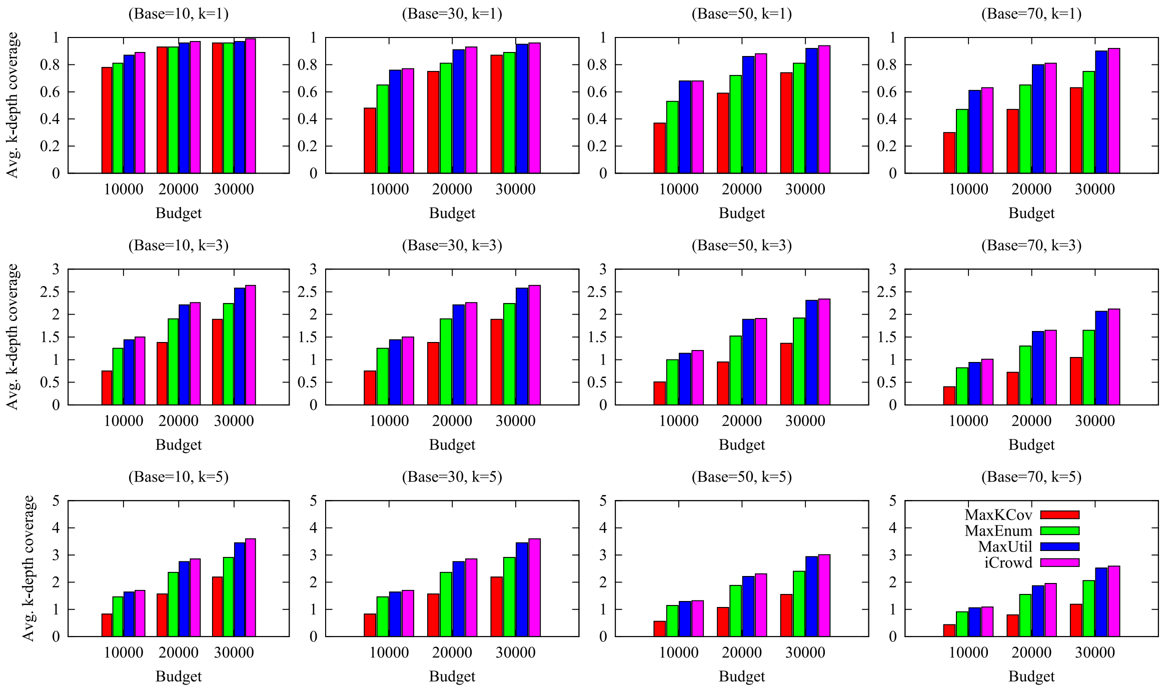


Fig. 3. k-depth Coverage Comparison in the MERGED region with varying individual incentive setting (Please see the Results of RESIDENTIAL and BUSINESS Regions in Appendix, available in the online supplemental material).

We used the four periods' call traces to simulate *four* PCS tasks, each lasting for two weeks. We assume that each PCS task executes five days per week. We carried out experiments using a laptop with an Intel Core i7-2630QM Quart-Core CPU and 8 GB memory. iCrowd and baseline algorithms were implemented with the Java SE platform on a Java HotSpot(TM) 64-Bit Server VM.

5.2 Baselines and Comparisons for Goal. 1

In order to evaluate iCrowd for Goal. 1, we first introduce three baselines derived from state of the art optimization algorithms, and then compare the performance of iCrowd to the baselines in terms of k-depth coverage achieved by iCrowd and three baselines under the same budget/incentive setting. Further we use a case study to illustrate the number of sensor readings collected from each subarea under the specific incentive/budget/k-depth setting.

Baselines for Goal. 1 –We provide three baseline task allocation methods using the greedy and partial enumeration for comparative studies.

- *MaxKCov* – This method adopts the same Greedy User-Cycle Combination Set search algorithm (*2a.* of iCrowd). A single-loop greedy search algorithm continues selecting/adding an unselected user-cycle combination $\langle u, i\rangle$ maximizing $kCov(\langle u, i\rangle \cup \mathbb{X})$ in each iteration until $cost(\langle u, i\rangle \cup \mathbb{X}) > B$, then returns $\mathbb{X}$ as task allocation result, where $\mathbb{X}$ refers to the user-cycle combinations already selected in previous iterations and $cost(\mathbb{X})$ is the budget consumption of the selected user-cycle combinations.
- *MaxUtils* – This method uses the same Greedy User-Cycle Combination Set search algorithm as MaxKCov but with a different utility function $\frac{kCov(\langle u,i\rangle\cup\mathbb{X})-kCov(\mathbb{X})}{cost(\langle u,i\rangle\cup\mathbb{X})-cost(\mathbb{X})}$ for maximization.
- *MaxEnum* –Rather than selecting a user-cycle combination in each iteration, this greedy algorithm first enumerates all possible cycle subsets of each unselected user, and selects an unselected user and his/her cycle subset $\mathbb{C}_v^{\#}$ having the maximal utility $\frac{kCov(\mathbb{C}_v^{\#}\cup\mathbb{X})-kCov(\mathbb{X})}{b_a+b_o*|\mathbb{C}_v^{\#}|}$ in iteration; this algorithm continues adding an unselected user and his/her cycle subset in each iteration until $cost(\mathbb{C}_v^{\#} \cup \mathbb{X}) > B$, then returns $\mathbb{X}$ as the task allocation result. This algorithm is derived from [28].

Performance Comparisons for Goal. 1: k-depth Coverage Comparisons under the same Budget Constraint – In order to compare the performance of iCrowd to baselines for Goal. 1 in generalized settings, this experiment evaluates iCrowd and baselines under two sets of budget/incentive settings:

- **Varying Incentive Setting($B_o > 0$) –** In this setting, we aim to evaluate the performance of iCrowd and baselines, when the individual incentive payment to each participant, coverage depth, and overall budget are all varying. Specifically:
  - ○ The Bonus incentive is fixed to $b_o = 1$, while the base incentive is set to $B_a = 10, 30, 5, 070$;
  - ○ The total amount of incentive budget is set to $B =$10,000, 20,000 and 30,000;
  - ○ The k-depth Coverage threshold in each cell tower/sensing cycle is set to $k =$1, 3 and 5.

Fig. 3 presents the average k-depth Coverage in each cell tower and each sensing cycle (of the four PCS tasks) using the four methods, with the varying incentive setting.

- **Fixed Incentive Setting** ($B_o = 0$) – In this setting, we aim to evaluate the performance of iCrowd and baselines, when the individual incentive payment to each

TABLE 2
k-Depth Coverage Comparison in the BUSINESS Region with $b_o = 0$ and $b_a = 50$

k	iCrowd	MaxKCov	MaxEnum	MaxUtils
		$B = 10{,}000$		
1	0.83	0.49	0.83	0.79
3	1.73	0.71	1.73	1.52
5	2.0	0.79	2.0	1.74
		$B = 20{,}000$		
1	0.94	0.74	0.94	0.92
3	2.44	1.37	2.44	2.27
5	3.18	1.57	3.18	2.89
		$B = 30{,}000$		
1	0.96	0.85	0.96	0.95
3	2.72	1.89	2.72	2.62
5	3.97	2.16	3.97	3.67

participant is fixed while coverage depth and overall budget constraint varying. Specifically:

- Base/Bonus incentive payments are fixed to $B_a = 50$ and $B_o = 0$ respectively;
- The total amount of incentive budget is set to B =10,000, 20,000, and 30,000;
- The k-depth Coverage threshold in each cell tower/sensing cycle is set to k =1, 3 and 5.

Table 2 presents the average k-depth Coverage in each cell tower and each sensing cycle (of the four PCS tasks) using the four methods, with the fixed incentive setting.

Note that the average k-depth Coverage could not be bigger than the depth K, as the maximal k-depth Coverage of each cell tower/cycle is k. From the k-depth Coverage comparisons shown in Table 2 and Fig. 3, we can observe that *in all the cases iCrowd outperformed the three baselines under the same budget constraint*. Specifically, iCrowd achieved on average 60 percent higher k-depth Coverage than MaxKCov, 18 percent higher than MaxEnum, and 3 percent higher than MaxUtils. The evaluation results based on RESIDENTIAL region shows similar results. Note that when $B_O = 0$, iCrowd performs exactly the same as MaxEnum; these two algorithms select the same group of participants and assign sensing task to each cycle of each selected participant, as the search strategies utilized by these two algorithms are equivalent under the fixed incentive setting.

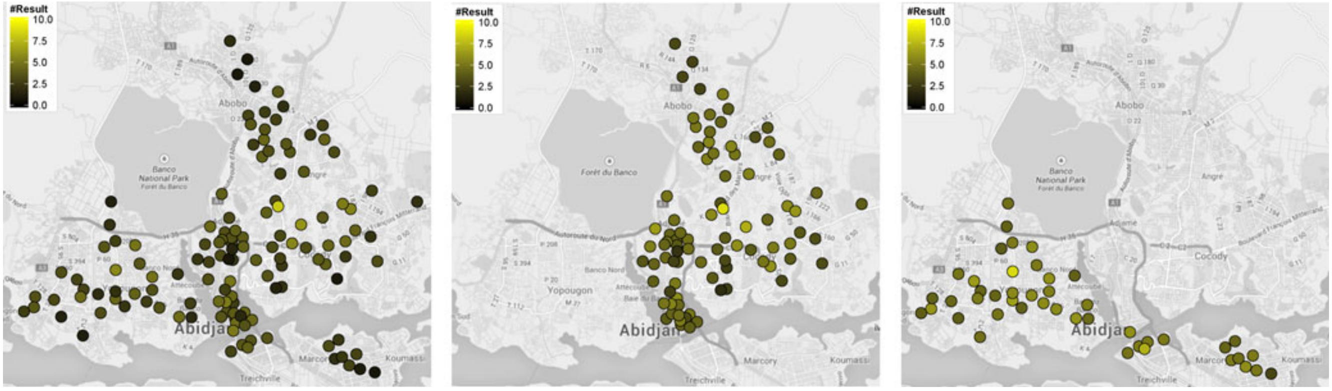
Case Study for Goal. 1: Spatial Distribution of the Sensor Readings – After evaluating the performance of iCrowd and three baselines from k-depth Coverage perspectives, we evaluate the spatial distribution of sensor readings using iCrowd with the target regions of different size. In Fig. 4, we present the average number of sensor readings returned from each cell tower in each sensing cycle using iCrowd, using the datasets from the BUSINESS, RESIDENTIAL and MERGED regions, with the same setting $B = 30,000$, $k = 5$, $b_a = 50$ and $b_o = 1$. From Fig. 4, we can see that *when using iCrowd, the sensor readings are uniformly distributed across cell towers in any of the three regions.* While the k-depth Coverage threshold in each cell tower/cycle is set to $k = 5$, the experiment shows that each cell tower gets on average 5.3, 4.5 and 3.3 sensor readings using datasets from RESIDENTIAL, BUSINESS and MERGED regions, respectively. Further the standard deviation is 0.98, 1.1 and 1.2 for three regions, respectively. This suggests that each cell tower can get a comparable number of sensor readings in any of the three regions using iCrowd.

5.3 Baselines and Comparisons for Goal. 2

In order to evaluate performance of iCrowd for Goal. 2, we build baselines and compare their performance to iCrowd, using following k-depth coverage constraint (k, r) and incentives (b_a, b_o) settings:

- The base and bonus incentives are fixed to $b_a = 1$ and $b_o = 0$ respectively;
- The k-depth coverage constraint (k, r) is set to coverage depth $k = 1$ and coverage ratio $r = 85\%$ and 95%.

In this case, the incentive payment to each participant is fixed at 1 and the overall incentive payment is equal to the number of selected participants. In terms of coverage, the depth is set to $k = 1$, thus the MCS task is targeted at collecting at least one sensor reading from at least a predefined percentage ($r = 85\%$ or 95%) of subareas in each sensing cycle. With these incentive and k-depth coverage settings, the objective of this evaluation is to select a minimal number of participants and allocate tasks to each sensing cycle of each selected participant, while ensuring the selected participants covering at least 85 or 95 percent subareas in each sensing cycle. We first introduce the baseline algorithms for



(a) RESIDENTIAL Region (b) BUSINESS Region (c) MERGED Region

Fig. 4. Spatial distribution of sensor readings in three regions (B=30,000, $k = 5$, $b_a = 50$ and $b_o = 1$).

TABLE 3
Overall Incentive Payment (i.e., #Selected Participants, where "iC." refers to iCrowd)

(a) **BUSINESS** Region

Task	iC.	MaxMin	MaxCom	MaxCov
		$r = 95\%$		
1	523	601	810	1309
2	510	576	822	1266
3	414	475	748	2130
4	704	753	1424	1965
avg.	537.8	601.3	951	1667.5
		$r = 85\%$		
1	261	289	413	525
2	257	274	390	454
3	198	255	379	455
4	318	354	506	822
avg.	258.5	293	422	564

(b) **RESIDENTIAL** Region

Task	iC.	MaxMin	MaxCom	MaxCov
		$r = 95\%$		
1	501	598	911	2112
2	512	638	714	2251
3	508	624	768	1701
4	500	631	631	1576
avg.	505.3	622.8	756	1910
		$r = 85\%$		
1	257	315	373	585
2	261	326	345	717
3	268	320	355	618
4	243	350	275	551
avg.	257.3	327.8	337	617.8

(c) **MERGED** Region

Task	iC.	MaxMin	MaxCom	MaxCov
		$r = 95\%$		
1	766	859	1239	2368
2	753	818	1113	2385
3	688	760	1109	2479
4	898	1012	1703	2537
avg.	776.3	862.3	1291	2442.3
		$r = 85\%$		
1	320	419	547	902
2	329	348	599	892
3	285	305	465	959
4	353	437	620	1236
avg.	321.8	377.3	557.8	997.3

Goal. 2 that can select a minimal number of participants while ensuring a predefined percentage (e.g., $r = 95\%$ or 85%) of subareas being covered by at least one sensing cycle (i.e., $k = 1$) in each sensing cycle, then we compare the overall incentive payment consumed by baselines and iCrowd under the same coverage setting.

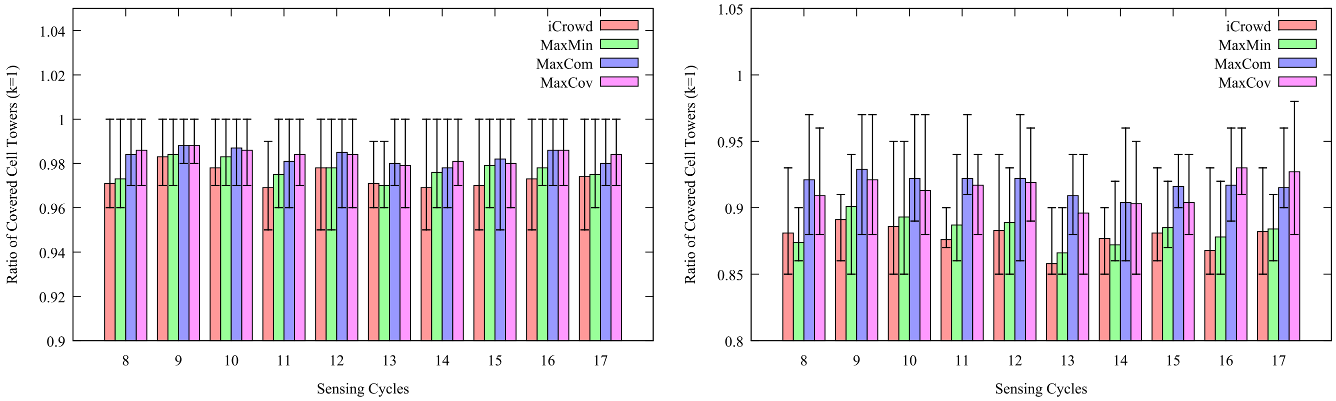
Baselines for Goal. 2 – In our evaluation, we provide three baseline methods with different utility-based selection strategy from iCrowd, but all of them share the same iteration process and stopping criterion.

1) *MaxMin* – MaxMin uses a single-loop greedy algorithm to select an unselected user and all his/her cycles in each iteration, until the k-depth coverage constraint achieved. Specifically, the algorithm, selects/adds a user having the maximal utility $min_{0 \leq i < I} P(\mathbb{X} \cup \{\langle u, i \rangle\}, k, r, i)$ in each iteration. Note that the utility function used here refers to the minimum of the coverage probabilities among all sensing cycles. Please refer to [29] for details.
2) *MaxCom* – The basic idea of MaxCom is to select the next participant who best complements with the selected set of participants in terms of coverage probability. This method is derived from the idea proposed by [21].
3) *MaxCov* – In each MaxCov is to simply select the next participant who covered the most cell towers in the historical call traces, among all the unselected mobile users.

In all experiments, we set the *stopping threshold* in *stopping criterion* using an empirical value of $P_{thr} = (99.99\%)^{1/(|T|*I)}$ for evaluating iCrowd as well as other three baselines.

Performance Comparisons for Goal. 2: Overall Incentive Payment Comparisons under the same k-depth coverage constraint - In Table 3 , we present the performance comparison on overall incentives consumption (i.e., number of selected participants for each of the four tasks) between iCrowd and baselines. It is clear that iCrowd outperformed MaxMin, MaxCom and MaxCov methods in all PCS tasks. On average, iCrowd consumed 10.0-21.5 percent less overall incentives compared to MaxMin (i.e., 10.0-21.5 percent fewer selected participants), consumed 23.7-43.5 percent less overall incentives compared to MaxCom, and consumed 54.2-73.5 percent less overall incentives compared to MaxCov. In terms of k-depth coverage, we show the Max/Min/Average the percentage of cell towers covered by at least one sensor reading (i.e., $k = 1$) in each sensing cycle in Fig. 5 using iCrowd and baselines. For all sensing cycles, the required coverage ratio (i.e., 95 and 85 percent) are achieved by the four methods without significant differences.

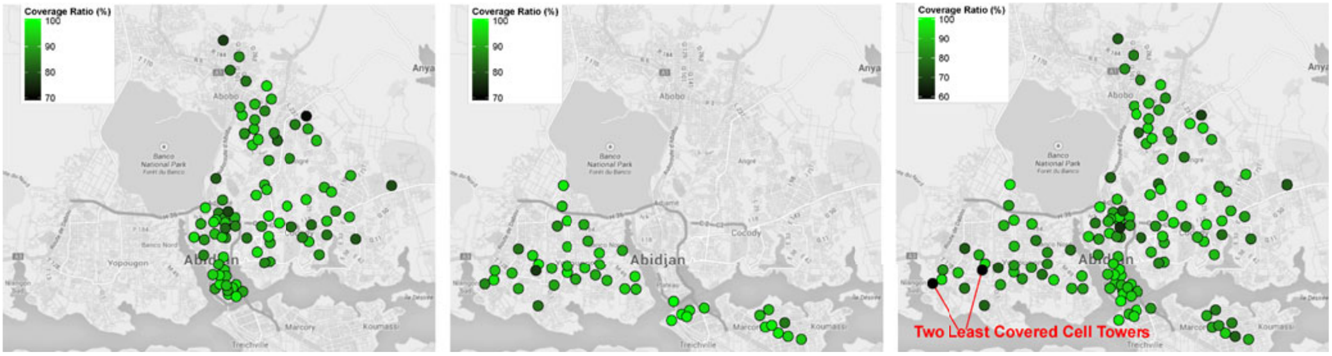
Case Study for Goal. 2: Temporal Coverage of Sensor Readings - As the coverage ratio r specified in this paper is less than 100 percent, it is conceivable that some cell towers may have low temporal coverage or zero coverage (e.g., no sensor readings in any sensing cycle). Thus we examined the temporal coverage of the cell towers using the three datasets when Ratio = 85%. As shown in Fig. 6, most cell towers can be covered in more than 80 percent sensing cycles when



(a) MERGED 95%

(b) MERGED 85%

Fig. 5. Max/Min/Average ratio of covered cell towers based on the MERGED region (Please see the result of RESIDENTIAL and BUSINESS regions in Appendix, available in the online supplemental material).



(a) BUSINESS 85% (b) RESIDENTIAL 85% (c) MERGED 85%

Fig. 6. Temporal coverage ratio of cell towers in BUSINESS, RESIDENTIAL, and MERGED regions.

using the BUSINESS, RESIDENTIAL and MERGED datasets. The two least covered cell towers (tower id = 724 and id = 646 in the MERGED region) were still covered in 59 percent of the sensing cycles.

Due to space limitation, some iCrowd evaluation results are not reported here. Readers are encouraged to see the Appendix, available in the online supplemental material, for additional details including the time consumption of iCrowd for both Goal. 1 and Goal. 2 and other experiment insights.

6 Conclusion

In this paper, we proposed a unified task allocation framework, *iCrowd*, for Piggyback Crowdsensing. iCrowd is designed to optimally allocate sensing tasks to PCS participants, subject to different incentive and spatial-temporal coverage constraints/objectives. Specifically, iCrowd could be adopted to either *maximize the overall k-depth coverage across all sensing cycles with a fixed budget* or to *minimize the overall incentive payment while ensuring a predefined k-depth coverage constraint*, by selecting a number of participants and determining in which sensing cycles each selected participant is needed for the PCS task participation. The PCS was adopted to reduce energy consumption of individual mobile device, by exploiting call opportunities to perform sensing tasks and upload sensed data. In order to allocate PCS task for either optimal MCS data collection goals, iCrowd first predicts the coverage probability of each mobile user, then performs a near-optimal participant/cycle task allocation search algorithm with low computational complexity. Theoretical analysis proves that iCrowd can achieve near-optimality for both optimal MCS data collection goals, and evaluations with a large-scale real-world dataset show that iCrowd outperformed six baseline approaches. For Goal. 1 it achieved 3-60 percent higher k-depth coverage compared to baseline approaches under the same budget constraint, while for Goal. 2 iCrowd required 10.0-73.5 percent less overall incentive compared to baselines under the same k-depth coverage constraint.

Acknowledgments

This work is the extension of our previous conference papers [1], [2]. This research is partly supported by the EU FP7 MONICA project (PIRSES-GA-2011-295222), the Hobby Postdoctoral and Predoctoral Fellowships in Computational Science, the NSFC grant No. 61572048, and the Microsoft collaborative research grant. D. Zhang is the corresponding author.

References

[1] D. Zhang, H. Xiong, L. Wang, and G. Chen, “Crowdrecruiter: Selecting participants for piggyback crowdsensing under probabilistic coverage constraint,” in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2014, pp. 703–714.

[2] H. Xiong, D. Zhang, G. Chen, L. Wang, and V. Gauthier, “Crowdtasker: Maximizing coverage quality in piggyback crowdsensing under budget constraint,” in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, 2015, pp. 55–62.

[3] R.K. Ganti, F. Ye, and H. Lei, “Mobile crowdsensing: Current state and future challenges,” *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 32–39, Nov. 2011.

[4] D. Zhang, L. Wang, H. Xiong, and B. Guo, “4w1h in mobile crowd sensing,” *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 42–48, Aug. 2014.

[5] Y. Zheng, F. Liu, and H.-P. Hsieh, “U-air: When urban air quality inference meets big data,” in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 1436–1444.

[6] N. Roy, A. Misra, C. Julien, S. K. Das, and J. Biswas, “An energy-efficient quality adaptive framework for multi-modal sensor context recognition,” in *Proc. IEEE Conf. Pervasive Comput. Commun.*, 2011, pp. 63–73.

[7] M. Musolesi, M. Piraccini, K. Fodor, A. Corradi, and A.T Campbell, “Supporting energy-efficient uploading strategies for continuous sensing applications on mobile phones,” *Pervasive Comput.*, 2010, pp. 355–372.

[8] N. D. Lane, Y. Chon, L. Zhou, Y. Zhang, F. Li, D. Kim, G. Ding, F. Zhao, and H. Cha, “Piggyback crowdsensing (pcs): Energy efficient crowdsourcing of mobile sensor data by exploiting smartphone app opportunities,” in *Proc. 11th ACM Conf. Embedded Netw. Sens. Syst*, 2013, p. 7.

[9] L. Wang, D. Zhang, and H. Xiong, “Effsense: Energy-efficient and cost-effective data uploading in mobile crowdsensing,” in *Proc. ACM Conf. Pervasive Ubiquitous Comput. Adjunct Publication*, 2013, pp. 1075–1086.

[10] H. Xiong, D. Zhang, L. Wang, and H. Chaouchi, “Emc3: Energy-efficient data transfer in mobile crowdsensing under full coverage constraint,” *IEEE Trans. Mobile Comput.*, vol. 14, no. 7, pp. 1355–1368, Jul. 2015

[11] X. sheng, J. Tang, and W. Zhang, “Energy-efficient collaborative sensing with mobile phones,” in *Proc. IEEE Conf. Comput. Commun.*, 2012, pp. 1916–1924.

[12] A. Ahmed, K. Yasumoto, Y. Yamauchi, and M. Ito, “Distance and time based node selection for probabilistic coverage in people-centric sensing,” in *Proc. IEEE Conf. Sens., Commun. Netw.*, 2011, pp. 134–142.

[13] D. Zhang, H. Xiong, L. Wang, and G. Chen, “Crowdrecruiter: Selecting participants for piggyback crowdsensing under probabilistic coverage constraint,” in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2014, pp. 703–714.

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

[15] S. Hachem, A. Pathak, and V. Issarny, "Probabilistic registration for large-scale mobile participatory sensing," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, 2013, vol. 18, p. 22.

[16] F. Murena, "Measuring air quality over large urban areas: Development and application of an air pollution index at the urban area of naples," *Atmospheric Environ.*, vol. 38, no. 36, pp. 6195–6202, 2004.

[17] US Environmental Protection Agency, "Guidance on choosing a sampling design for environmental data collection—for use in developing a quality assurance project plan (epa qa/g-5s)," 2002.

[18] C. Xu, S. Li, G. Liu, Y. Zhang, E. Miluzzo, Y-F Chen, J. Li, and B. Firner, "Crowd++: Unsupervised speaker count with smartphones," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2013, pp. 43–52.

[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," *CoRR*, abs/1210.0137, 2012.

[20] S. Reddy, K. Shilton, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Using context annotated mobility profiles to recruit data collectors in participatory sensing," in *Location and Context Awareness*. New York, NY, USA: Springer, 2009, pp. 52–69.

[21] S. Reddy, D. Estrin, and M. Srivastava,, "Recruitment framework for participatory sensing data collections," in *Proc. 8th Int. Conf. Pervasive*, 2010, pp. pp. 138–155.

[22] A. Singla and A. Krause, "Incentives for privacy tradeoff in community sensing," in *Proc. 1st AAAI Conf. Human Comput. Crowdsourcing*, 2013, pp. 165–173.

[23] 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 Commun. Mag.*, vol. 51, no. 6, pp. 112–119, Jun. 2013.

[24] D. Philipp, J. Stachowiak, P. Alt, F. Dürr, and K. Rothermel, "Drops: Model-driven optimization for public sensing systems," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, 2013, vol. 18, p. 22.

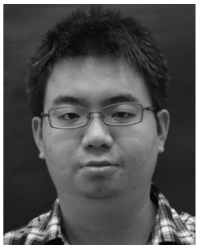
[25] J. Weinberg, L. D. Brown, and J. R. Stroud, "Bayesian forecasting of an inhomogeneous poisson process with applications to call center data," *J. Am. Statist. Assoc.*, vol. 102, no. 480, pp. 1185–1198, 2007.

[26] C. W. Gardiner, et al.., *Handbook of Stochastic Methods*, vol. 3. Berlin, Germany: Springer, 1985.

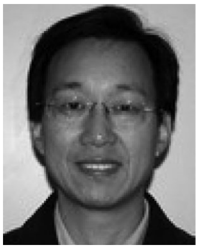
[27] R. K. Iyer and J. A. Bilmes, "Submodular optimization with submodular cover and submodular knapsack constraints," in *Proc. Ad. Neural Inf. Process. Syst.*, 2013, pp. 2436–2444.

[28] M. Sviridenko, "A note on maximizing a submodular set function subject to a knapsack constraint," *Oper. Res. Lett.*, vol. 32, no. 1, pp. 41–43, 2004.

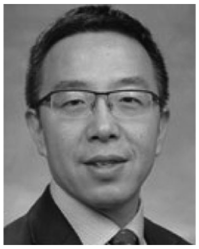
[29] E. L. Ulungu and J. Teghem, "Multi-objective combinatorial optimization problems: A survey," *J. Multi-Criteria Decision Anal.*, vol. 3, no. 2, pp. 83–104, 1994.



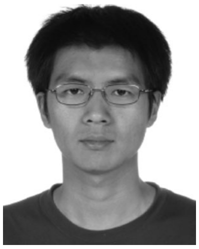
Haoyi Xiong received the PhD degree in computer science from the Institut Mines-Télécom, TELECOM SudParis and Université Pierre et Marie Curie, France. He is currently a research associate in the Department of Systems and Information Engineering, University of Virginia, VA. His research interests include mobile crowdsensing and ubiquitous computing. He received the Best Paper Award from IEEE UIC 2012, and is currently a member of the IEEE.



Daqing Zhang received the PhD degree from the University of Rome "La Sapienza" and the University of L'Aquila, Italy, in 1996. He is a professor at the Institut Mines-Télécom, TELECOM SudParis, France. His research interests include large-scale data mining, urban computing, context-aware computing, and ambient assistive living. He is an associate editor for four journals including the *ACM Transactions on Intelligent Systems and Technology* and *IEEE Transactions on Big Data*. He is the winner of the 10 Years CoMoRea Impact Paper Award at IEEE PerCom 2013, the Best Paper Award at IEEE UIC 2012, and the Best Paper Runner Up Award at Mobiquitous 2011.



Guanling Chen received the BS degree in computer science from Nanjing University in 1997, and the PhD degree in computer science from Dartmouth College in 2004. He is an associate professor of computer science at the University of Massachusetts Lowell. He is also an affiliate faculty in the Institute for Security, Technology, and Society (ISTS) at Dartmouth College. He was an I3P fellow before he joined the faculty of UMass Lowell in 2005. His research interests include mobile computing, ubiquitous computing, human-computer interaction, and intelligent systems. He has received more than $1.5 million research grants from US National Science Foundation (NSF) and the Department of Homeland Security (DHS).



Leye Wang received the BSc and MSc degrees in computer science from Peking University. He is currently working toward the PhD degree at the Institut Mines-Télécom, TELECOM SudParis, and Université Pierre et Marie Curie, France. His research interests include mobile crowdsensing and ubiquitous computing.

Vincent Gauthier received the PhD degree in electrical engineering and computer networks from the University of Paris 6 in 2006, respectively. He was a guest researcher at the National Institute of Standards and Technology, MA, from 2006 to 2008. He joined the faculty of Telecom SudParis and the Lab CNRS SAMOVAR (UMR 5157), Evry, France, in 2008, where he is currently an associate professor in the Department of Computer Networks. His current research interests are primarily in wireless networks and cooperative communications. His other research interests include mobility modeling, performance analysis, and queuing theory.

Laura E. Barnes received the PhD degree in computer science & engineering from the University of South Florida in 2008. She is an assistant professor in the Department of Systems and Information Engineering, at the University of Virginia. Prior to joining the University of Virginia, she held faculty positions at the University of South Florida in the College of Medicine and at the University of Texas Arlington in the College of Engineering. Her research interests include medical informatics, machine learning, mobile computing, and robotics.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.