The authors propose two alternating data collection goals. Goal 1 is maximizing overall spatial-temporal coverage under a predefined incentive budget constraint; goal 2 is minimizing total incentive payment while ensuring predefined spatial-temporal coverage for collected sensor data, all on top of the PCS task model.

**Abstract**

Piggyback crowdsensing (PCS) is a novel energy-efficient mobile crowdsensing paradigm that reduces the energy consumption of crowdsensing tasks by leveraging smartphone app opportunities (SAOs). This article, based on several fundamental assumptions of incentive payment for PCS task participation and spatial-temporal coverage assessment for collected sensor data, first proposes two alternating data collection goals. Goal 1 is maximizing overall spatial-temporal coverage under a predefined incentive budget constraint; goal 2 is minimizing total incentive payment while ensuring predefined spatial-temporal coverage for collected sensor data, all on top of the PCS task model. With all of the above assumptions, settings, and models, we introduce CrowdMind—a generic incentive allocation framework for the two optimal data collection goals, on top of the PCS model. We evaluated CrowdMind extensively using a large-scale real-world SAO dataset for the two incentive allocation problems. The results demonstrate that compared to baseline algorithms, CrowdMind achieves better spatial-temporal coverage under the same incentive budget constraint, while costing less in total incentive payments and ensuring the same spatial-temporal coverage, under various coverage/incentive settings. Further, a short theoretical analysis is presented to analyze the performance of CrowdMind in terms of the optimization with total incentive cost and overall spatial-temporal coverage objectives/constraints.

**Introduction**

With the rapid proliferation of sensor-equipped smartphones, mobile crowdsensing (MCS) [1] has become an efficient way to sense and collect environmental data of urban areas in real time (e.g., air quality, temperature, and noise level). Instead of deploying static and expensive sensor networks in urban areas, MCS leverages the sensors embedded in mobile phones and the mobility of mobile users to sense their surroundings, and utilizes the existing communication infrastructure (3G, WiFi, etc.) to collect data from mobile phones scattered in an 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.

Our earlier work [2] demonstrated that there are two main players in MCS: the Organizer, who is the person or organization coordinating the sensing task, and the Participants, who are the mobile users involved in the sensing task. An MCS task usually requires the organizer to recruit participants with incentive payment, to allocate sensing tasks to selected participants, and to collect sensor readings from these participants’ mobile devices that will represent the characteristics of the target sensing region, often with a predefined budget for 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., one week), 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 km² grid cells, then splits the one-week MCS process into a sequence of one-hour sensing cycles [3], and further requests at least one MCS participant in each grid to upload the air quality sensor reading in each sensing cycle. Besides the full spatial-temporal coverage [4], the organizer frequently uses the partial spatial-temporal coverage metrics for MCS data collection, where the fraction of subareas being covered by at least one sensor reading in each sensing cycle is used to represent the coverage [5]. 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 the majority part (e.g., 80 percent) of subareas in each cycle.

In addition to organizers’ efforts in the process of participant recruitment, task assignment, and data collection, MCS also requires the participants’ mobile devices to sustainably perform sensing tasks and upload sensor data during the MCS process. In order to prolong the battery life of mobile devices engaged in MCS, various solutions have been proposed to reduce energy consumption of individual mobile devices, ranging from adapting sensing frequency to inferring part of data rather than sensing and uploading all data [6]. One of the effective solutions is piggyback
crowdsensing (PCS), which reduces energy consumption by leveraging smartphone opportunities [7]. Generally, a PCS app works as follows.

Before the MCS process, the organizer needs to first recruit participants and assign each recruited participant a list of sensing cycles that he/she needs to return sensor data. Furthermore, it requires, as shown in Fig. 1, that each participant’s MCS device performs a sensing task and returns sensor readings immediately when the smartphone SAO is available in the sensing cycles with the PCS task assigned. By performing sensing tasks and uploading sensor data in parallel with an SAO, the PCS task model can significantly reduce energy consumption caused by crowdsensing [7]. 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 [8]. The work described in this article is based on our very recent publication [9]; we focus on the big picture, framework, and general idea in this article, while more technical details about the crowd sensing algorithm and its analysis can be found there. These two articles together serve as a whole piece, reporting our recent progress in incentive task allocation for crowdsensing.

To incentivize participants in mobile crowdsensing, the organizer usually pays each recruited participant a constant amount as the base incentives; then a varying amount of bonus incentives would be offered depending on the number of cycles assigned with PCS tasks.

To incentivize participants in mobile crowdsensing, the organizer usually pays each recruited participant a constant amount as the base incentives, then a varying amount of bonus incentives would be offered depending on the number of cycles assigned with PCS tasks.

To incentivize participants in mobile crowdsensing, the organizer usually pays each recruited participant a constant amount as the base incentives, then a varying amount of bonus incentives would be offered depending on the number of cycles assigned with PCS tasks.

### Key Challenges

- Maximizing the spatial-temporal coverage
- Minimizing the total budget while ensuring the spatial-temporal coverage

### Problem Formulation and Key Challenges

In this section, we first provide an overview of the general incentive allocation problem that unifies incentive task allocation problems under various incentives, spatial-temporal coverage, and data collection settings. Based on the generic problem formulation, we elaborate several key technical challenges.

Here, we first define the overall set of participants as \( U = \{u_0, u_1, ..., u_n\} \), where each \( u_i \in U \) refers to a participant, the set of sensing cycles for PCS task as \( t = \{t_0, t_1, ..., t_m\} \), where each \( t_j \in T \) refers to a sensing cycle, and the task assignment as \( \alpha = \{(u_i, t_j) \} \subset U \times T \) where each \((u_i, t_j) \in \alpha\) refers to assigning a PCS task to \( U_i \) in sensing cycle \( t_j \); then the set of sub-regions in the target area as \( C = \{c_0, c_1, ... c_k\} \). We define the sub-regions that are covered by the task assignment \( \alpha \) at sensing cycle \( t \) as \( \text{cover}(\alpha) \), where \( \text{cover}(\alpha) \subseteq C \).

Further, we define \( U(\alpha) \) as the number of unique mobile users assigned with PCS tasks in \( \alpha \). Thus, the overall incentive budget consumption should be \( b_\alpha \cdot U(\alpha) + b_\alpha^* \cdot |\alpha| \), which considers both base payment for each recruited participant and bonus incentive for each assigned task.

**General Incentive Allocation Problem:** With all of the above definitions in mind, the general form of the task allocation problem is to find \( \alpha \) such that

\[
\text{For Goal.1: } \alpha = \arg \max_{\alpha'} \sum_{\alpha \in T} |\text{cover}(\alpha')| \\
\text{s.t. } b_\alpha \cdot U(\alpha) + b_\alpha^* \cdot |\alpha| \leq B \\
\text{For Goal.2: } \alpha = \arg \max_{\alpha'} b_\alpha \cdot U(\alpha) + b_\alpha^* \cdot |\alpha| \\
\text{s.t. } |\text{cover}(\alpha')| \geq G, \forall \alpha \in T
\]

where \( B \) is the predefined budget for data collection. Goal.1 and \( G \) refer to the expected number of covered sub-regions for Goal.2 (\( G < |C| \) for partial coverage and \( G = |C| \) for the full coverage), respectively. Note that \( \text{cover}(\alpha) \) depends on the future mobility and SAOs of the selected participants/cycles in \( \alpha \), which is not known beforehand. Our work intends to solve the proposed problems through stochastic optimization and SAO/mobility prediction.

To solve the above problems, we have to tackle the following technical challenges:

- **Predicting the future mobility and SAO of crowds using the historical mobility/SAO traces:** To allocate incentive for SAO-based PCS tasks, we might first need to predict each participant’s mobility and SAOs during the entire PCS period, based on their historical mobility/SAO traces. In this way, we can predict when and where each participant will be more likely to return a sensor reading if a task is assigned.

- **Estimating the Spatial-Temporal Coverage Using the Predicted Mobility and SAOs:** Given the predicted mobility/SAOs of all participants and the task allocation result \( \alpha \), we then need to estimate the likelihood of each subarea being covered by at least one sensor reading in each sensing cycle. To estimate the spatial-temporal coverage for data collection Goal.1, we can easily sum the likelihood of every subarea being
covered in every sensing cycle to calculate the expectation of $\Sigma_t |\text{cover}(a)|$. On the other hand, to test whether the predefined spatial-temporal coverage is achieved (i.e., $\text{cover}(a) \geq G$), we also need a method to estimate the lower bound of $|\text{cover}(a)|$, $\forall t \in T$.

**Optimal Task Assignment Using the Estimated Coverage**

Given the function that can estimate $|\text{cover}(a)|$ for any task assignment $a$ using predicted mobility/SAO, there further needs to be a method to search the optimal $a$. In the target problem, the optimal solution should select a subset of participants from all users; then, for each selected participant, assign PCS tasks to a subset of her sensing cycles, with respect to the data collection goal. Thus, combinatorial optimization is needed to search among all the $2^{U \times T}$ choices to solve the optimization problem for Goal.1 or Goal.2. Despite the NP-hardness of such combinatorial search, certain polynomial-time approximation algorithms are required to lower the computational complexity of the (near)-optimal search for $a$. 

**CROWDMIND: A GENERIC FRAMEWORK FOR PCS INCENTIVE ALLOCATION**

With all the above technical challenges in mind, we propose CROWDMIND — A Generic Framework for PCS Incentive Allocation. CROWDMIND includes a set of mobility prediction, coverage estimation, and near-optimal task allocation algorithms, which could achieve the near-optimal incentive allocation for both data collection goals under certain incentive/coverage assumptions.

**MOBILITY PREDICTION USING MOBILITY TRACES**

Assuming the SAO sequence follows an inhomogeneous spatial-temporal Poisson process, the probability of a user $U_t$ to have at least one SAO at subarea $c_j$ ($c_j \in C$) in sensing cycle $t \in T$ can be calculated as $P_{\text{U}(c_j,t)} = 1 - e^{-\lambda(c_j)t}$, where $\lambda(c_j)$ refers to the Poisson intensity, which is estimated as the average number of SAOs that $U_t$ has placed at $c_j$ in the historical traces corresponding to the sensing cycle $t$. For example, to estimate $\lambda(c_j)$ for sensing cycle $t$ from 08:00 to 09:00, we count the average number of SAOs placed by $U_t$ at $c_j$ during the same period 08:00–09:00 in historical traces.

**COVERAGE ESTIMATION AND PROBABILISTIC LOWER BOUND**

For Goal.1, given the task assignment $a$ and SAO/mobility prediction result $P_{\text{U}(c_j,t)}$, we estimate the spatial-temporal coverage for Goal.1 as the expectation of $\Sigma_{t \in T} |\text{cover}(a)|$, i.e., $E[\Sigma_{t \in T} |\text{cover}(a)|] = \Sigma_{t \in T} E[|\text{cover}(a)|]$. We estimate the probability of the subarea $c_j$ being covered by $a$ in sensing cycle $t$ as: $P_{c_j(a,t)} = 1 - \prod_{\text{area}} (1 - P_{\text{U}(c_j,t)})$ is the probability of the subarea $c_j$ being covered by $a$ in sensing cycle $t$, and $U_t(a) \subseteq U$ refers to the set of participants assigned with task in sensing cycle $t$.

For Goal.2, we need to estimate if at least $G$ subareas were covered by assigned tasks in $a$. Thus, we calculate the probability of at least $G$ subareas being covered by $a$ in sensing cycle $t$ (i.e., probabilistic lower bound) as $P[|\text{cover}(a)| \geq G] = \sum_{G \subseteq \text{goal}} \sum_{a \subseteq \text{area}} \prod_{c_j \in G} P_{c_j(a)} \prod_{c_j \notin G} (1 - P_{c_j(a)})$.

To solve Eq. 2, we implemented a low-complexity algorithm using the second-moment generating function [10].

**SUBMODULAR OPTIMIZATION FOR TASK ASSIGNMENT**

We leverage submodular maximization algorithms to solve the combinatorial optimization problems of the PCS incentive allocation. We first introduce the algorithms used for Goal.1, then extend to Goal.2.

For Goal.1, the problem can be transformed to finding $a$ that maximizes $E[|\text{cover}(a)|]$ subject to $b_2^**(a) + b_3^*|a| \leq B$. As $E[\Sigma_{t \in T} |\text{cover}(a)|]$ is a monotonic, non-decreasing, submodular function, this problem could be solved by state-of-the-art constrained submodular maximization algorithms, as described below.

When $b_2 = 0$ and $b_3 > 0$: the problem becomes a submodular set function maximization over a knapsack constraint problem. In this case, a simple incremental greedy algorithm [11], which selects the user-cycle pair $(u, c)$ having the maximal spatial-temporal coverage increment, that is, $(\Delta E(|\text{cover}(a) \cup \{u, c\}) - E(|\text{cover}(a)|))/b_3$, could achieve the near-optimal incentive allocation. We first introduce the algorithms for Goal.1, which use budget feasibility as the stopping criterion of greedy
search, the greedy algorithms used for Goal.2 leverage probabilistic lower bound Eq. 2 as the stopping criterion. Specifically, the greedy search process here continues selecting/adding new participant-cycle pairs into \( a_i \), until \( P[\text{cover}_i(a_i) \geq G] \geq \theta_r \), where \( 0 < \theta_r < 1 \) is a predefined threshold to bound the spatial-temporal coverage. According to the Markov inequality, we have:

\[
P[\text{cover}_i(a_i) \geq G] \leq \frac{E[\text{cover}_i(a_i)]}{G}.
\]  

(3)

The proposed greedy search indeed optimizes the upper bound of Eq. 2 in each iteration and stops when Eq. 2 achieves the predefined threshold. We can conclude that, for Goal.2, the greedy algorithms are near-optimal under the optimization assumptions.

**Evaluation and Result Analysis**

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

**Dataset and Experiment Setups**

The dataset used in evaluation is the D4D dataset [13], which contains 50,000 users’ phone call traces (each call records includes user ID, call time, and cell tower) from Cote d’Ivoire. We use this phone call dataset for the evaluation, with the following assumption:

- **Assumption**: We consider each phone call placed by these users is an SAO for the potential sensor reading uploading.

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 have been installed and around 7945–8799 mobile phone users place phone calls in any two-week period
- **RESIDENTIAL**: a residential area where 45 cell towers have been installed and around 6034–6627 mobile phone users place phone calls in any two-week period
- **BUSINESS+RESIDENTIAL**: a combined area of both BUSINESS and RESIDENTIAL regions where 131 cell towers have been installed and around 11,363–12,049 unique mobile phone users place phone calls in any two-week slot.

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 Quad-Core CPU and 8 GB memory. CrowdMind and baseline algorithms were implemented with the Java SE platform on a Java HotSpot™ 64-Bit Server VM.

**Baselines and Comparisons for Goal.1**

In order to evaluate CrowdMind for Goal.1, we first introduce three baselines derived from state-of-the-art optimization algorithms, and then compare the performance of CrowdMind to the baselines in terms of coverage achieved by CrowdMind 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/setting.

**Baselines for Goal.1**: We provide three baseline task allocation methods using the greedy and partial enumeration schemes for comparative studies: MaxCov — adding a user-cycle pair that maximizes coverage in each iteration without considering the incentive cost [8]; MaxUtil — adding a user-cycle pair that has the maximal ratio of coverage improvement vs. incentive costs in each iteration; and MaxEnum — adding a user and a combination of his/her cycles that have the maximal ratio of coverage improvement vs. incentive costs; the algorithm is derived from [14].

**Performance Comparisons for Goal.1**: Spatial-Temporal Coverage Comparisons under the Same Budget Constraint: From the spatial-temporal coverage comparisons shown in Fig. 3, we can observe that in all the cases CrowdMind outperformed the three baselines with the same budget constraint, under the incentive and budget settings: \( b_a = 10/30/50/70 \), \( b_o = 1 \), and \( b = 1000/2000/3000 \). In the case of \( b_o = 0 \), we illustrate the average spatial-temporal coverage comparison of the four methods based on the BUSINESS region with various budgets in Table 1. Note that the average spatial-temporal coverage could not be bigger than 100 percent, that is,

\[
\sum_{a \in T} \frac{\text{cover}_a(a)}{|T| |C|} \leq 100\%.
\]
In our evaluation, we (30,000 0.94 30,000 30,000 using an 1/(| 30,000): In Table 2, we present the 10,000 10,000 10,000 10,000: In this article, we propose a unified incentive allo-

<table>
<thead>
<tr>
<th>CrowdMind</th>
<th>MaxCov</th>
<th>MaxEnum</th>
<th>MaxUtil</th>
</tr>
</thead>
<tbody>
<tr>
<td>B = 10000</td>
<td>0.83</td>
<td>0.49</td>
<td>0.83</td>
</tr>
<tr>
<td>B = 20000</td>
<td>0.94</td>
<td>0.74</td>
<td>0.94</td>
</tr>
<tr>
<td>B = 30000</td>
<td>0.96</td>
<td>0.85</td>
<td>0.96</td>
</tr>
</tbody>
</table>

Table 1. Average spatial-temporal coverage comparison in the BUSINESS region with $b_a = 0$ and $b_s = 50$.

**Baselines and Comparisons for Goal.2**

In order to evaluate performance of CrowdMind for Goal.2, we build baselines and compare their performance to CrowdMind using the following spatial-temporal coverage constraint and incentive ($b_a, b_s$) settings:

- The base and bonus incentives are fixed to $b_s = 1$ and $b_a = 0$ respectively.
- The spatial-temporal coverage constraint is set to the coverage ratio

$$G = \frac{C}{|C|} = 85\% \text{ and } 95\%.$$  

**Baselines for Goal.2:** In our evaluation, we provide three baseline methods with different utility-based selection strategies from CrowdMind, but all of them share the same iteration process and stopping criterion. The baselines are: MaxMin — instead of using the expectation of spatial-temporal coverage, this method using

$$\min_{r \in R} \{P(\text{cover}(\alpha) \geq G)\}$$

as the utility function of maximization; MaxCom — this method is derived from the idea proposed by [15], which selects the most “complementary” user-cycle pair in each iteration; and MaxCov — this method uses the same utility function as MaxCov for Goal.1. In all experiments, we set the stopping threshold in stopping criterion using an empirical value of $\text{thr} = (99.99\% \text{ percent})^{1/(C(T+T))}$ for evaluating CrowdMind as well as the other three baselines.

**Performance Comparisons for Goal.2:** Overall Incentive Payment Comparisons under the Same Coverage Constraint: In Table 2, we present the performance comparison on overall incentives consumption (i.e., number of selected participants for each of the four tasks) between CrowdMind and baselines. It is clear that CrowdMind outperformed the MaxMin, MaxCom, and MaxCov methods in all PCS tasks. On average, CrowdMind 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. All these methods meet the predefined coverage constraints.

**Conclusion and Discussion** In this article, we propose a unified incentive allocation framework, CrowdMind, for piggyback crowdsensing. CrowdMind is designed to opti-
nally allocate sensing tasks to PCS participants, subject to different incentive and spatial-temporal coverage constraints/objectives. Theoretical analysis proves that CrowdMind can achieve near-optimality for the two optimal MCS data collection goals, and evaluations with a large-scale real-world dataset show that CrowdMind outperformed all other baseline algorithms.

ACKNOWLEDGMENTS
The authors would like to thank the reviewers and the editors for their helpful comments and suggestions. Correspondence should be sent to D. Zhang (daqing.zhang@telecom-sudparis.eu) or H. Xiong (xiongha@mst.edu). This article was partially supported by the Startup Grant of Missouri S&T CPS Cluster Hire, the UVA Hobby Postdoctoral Fellowship for Computational Sciences, and the NSFC under Grant No. 61572048.

REFERENCES

BIographies
HAOYI XIONG is an assistant professor in the Department of Computer Science at Missouri University of Science and Technology. He obtained his B.E. degree in computer science and technology from Tsinghua University, Beijing, China, in 2009, his M.Phil. degree in mechanical and automation engineering from the Chinese University of Hong Kong in 2011, and his Ph.D. degree in computer science at the University of North Carolina from the University of North Carolina at Chapel Hill in 2016. His research and teaching interests include real-time scheduling, cyber-physical systems, and neural networks and their applications.

GUANING CHEN is an associate professor of computer science at the University of Massachusetts Lowell. After receiving his B.S. from Nanjing University, he completed his Ph.D. at Dartmouth College in 2004. His research interests include human-computer interaction and ubiquitous computing. He has published over 80 papers and received the Best Paper Award at MobileHCI 2009. He served as the TPC Chair of IEEE UIC 2017 and a Guest Editor of Sensors and the Journal of Healthcare Engineering.

LAURA E. BARNES (lbs3dp@virginia.edu) is an assistant professor in the Department of Systems and Information Engineering at the University of Virginia. She directs the Sensing Systems for Health Lab. Her research interests include the design, development, and evaluation of human-centered technologies to improve health and well-being. Barnes received her Ph.D. in computer science from the University of South Florida.

Table 2. Average incentive payment (b_2 = 0 and b_3 = 1, where “CM.” refers to CrowdMind).

<table>
<thead>
<tr>
<th>(a) BUSINESS region</th>
<th>(b) RESIDENTIAL region</th>
<th>(c) RESIDENTIAL+BUSINESS region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CM.</td>
<td>MaxMin</td>
<td>MaxCom</td>
</tr>
<tr>
<td>5578</td>
<td>601.3</td>
<td>951</td>
</tr>
<tr>
<td>258.5</td>
<td>293</td>
<td>422</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(b) RESIDENTIAL region</th>
<th>(c) RESIDENTIAL+BUSINESS region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CM.</td>
<td>MaxMin</td>
</tr>
<tr>
<td>5578</td>
<td>601.3</td>
</tr>
<tr>
<td>258.5</td>
<td>293</td>
</tr>
</tbody>
</table>