Near-Optimal Incentive Allocation for Piggyback Crowdsensing

Haoyi Xiong, Daqing Zhang, Zhishan Guo, Guanling Chen, and Laura E. Barnes

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 km^2 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

Digital Object Identifier: 10.1109/MCOM.2017.1600748

Haoyi Xiong and Zhishan Guo are with Missouri University of Science and Technology; Daqing Zhang is with Peking University; Guanling Chen is with the University of Massachusetts Lowell;; Laura E. Barnes is with the University of Virginia.

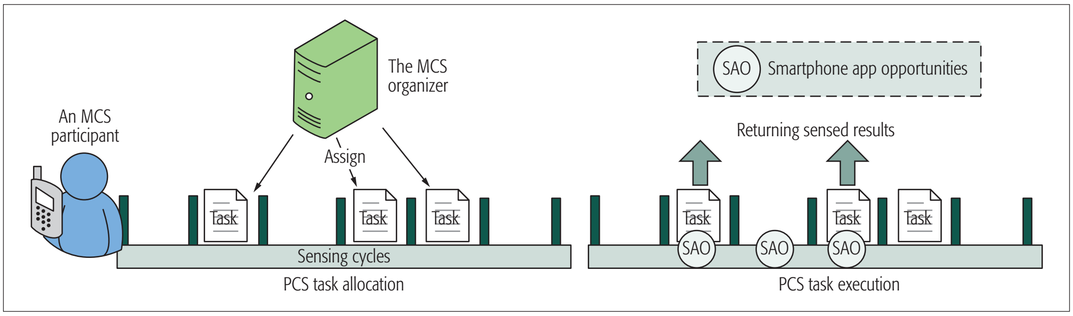

Figure 1. Task assignment for piggyback crowdsensing.

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. Once the spatial-temporal coverage and incentive payment settings are determined, the MCS organizer needs to recruit participants and assign PCS tasks with *either* of the following two data collection goals:

- *Maximizing* the spatial-temporal coverage with a *fixed budget*
- *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 $U = \{u_0, u_1, \ldots u_n\}$ where each $u_i \in U$ refers to a participant, the set of sensing cycles for PCS task as $t = \{t_0, t_1, \ldots t_m\}$, where each $t_j \in T$ refers to a sensing cycle, and the task assignment as $\alpha = \{(u_i, t_j) \ldots \subseteq U \times T$ 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 $C = \{c_0, c_1, \ldots c_k\}$. We define the sub-regions that are covered by the task assignment α at sensing cycle t as $cover_t(\alpha)$, where $cover_t(\alpha) \subseteq C$. Further, we define $U(\alpha)$as the number of unique mobile users assigned with PCS tasks in α. Thus, the overall incentive budget consumption should be $b_a{}^*u(\alpha')+b_o{}^*|\alpha'|$, 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 α such that

$$\begin{aligned}
&\text{For Goal.1}: \alpha = \underset{\alpha'}{\arg\max} \sum_{t \in T} |cover_t(\alpha')| \\
&\qquad s.t.\, b_a * u(\alpha') + b_o * |\alpha'| \le B \\
&\text{For Goal.2}: \alpha = \underset{\alpha'}{\arg\max}\, b_a * u(\alpha') + b_o * |\alpha'| \\
&\qquad s.t. |cover_t(\alpha')| \ge G, \forall t \in T
\end{aligned} \tag{1}$$

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 $cover_t(\alpha)$ depends on the future mobility and SAOs of the selected participants/cycles in α, 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 α, 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 eas-

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.

ily sum the likelihood of every subarea being covered in every sensing cycle to calculate the expectation of $\Sigma_t|cover_t(\alpha)|$. On the other hand, to test whether the predefined spatial-temporal coverage is achieved (i.e., $cover_t(\alpha) \geq G$), we also need a method to estimate the lower bound of $|cover_t(\alpha)|$, $\forall t \in T$.

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.

Optimal Task Assignment Using the Estimated Coverage: Given the function that can estimate $|cover_t(\alpha)|$ for any task assignment α using predicted mobility/SAO, there further needs to be a method to search the optimal α.

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 α.

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_i to have at least one SAO at subarea c_j ($c_j \in C$) in sensing cycle $t \in T$. This can be can be calculated as $P_{ui,cj,t} = 1 - e^{-\lambda ui,cj,t}$, where $\lambda_{ui,cj,t}$ refers to the Poisson intensity, which is estimated as the average number of SAOs that U_i has placed at c_j in the historical traces corresponding to the sensing cycle t. For example, to estimate $\lambda_{ui,cj,t}$ for sensing cycle t from 08:00 to 09:00, we count the average number of SAOs placed by U_i 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 α and SAO/mobility prediction result $P_{ui,cj,t}$, we estimate the spatial-temporal coverage for Goal.1 as the expectation of $\Sigma_{t\in T}|cover_t(\alpha)|$, i.e., $E(\Sigma_{t\in T}|cover_t(\alpha)|) = \Sigma_{t\in T}\Sigma_{c_j\in C}P_{c_j,t}(\alpha)$. We estimate the probability of the subarea c_j being covered by α in sensing cycle t as: where $P_{c_j,t}(\alpha) = 1 - \prod_{\forall ui \in ut}(\alpha(1 - P_{ui,cj,t}$ is the probability of the subarea c_j being covered by α in sensing cycle t, and $U_t(\alpha) \subseteq U$ 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 would be covered by assigned tasks in α. Thus, we calculate the probability of at least G subareas being covered by α in sensing cycle t (i.e., probabilistic lower bound) as

$$P\left(|cover_t(\alpha)| \geq G\right) = \sum_{G\leq g\leq|C|}\sum_{c\leq C,|c|=g}\prod_{c_j\in c}P_{c_c,t}(\alpha)\prod_{c_j\in C\backslash c}\left(1-P_{c_c,t}(\alpha)\right) \tag{2}$$

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 α that maximizes $E(\Sigma_t|cover_t(\alpha)|)$subject to $b_a{*}u(\alpha) + b_o{*}|\alpha| \leq B$. As $E(\Sigma_{t\in T}|cover_t(\alpha)|)$ 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 $b_a = 0$ and $b_o > 0$: 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_i, c_j) having the maximal spatial-temporal coverage increment, that is, $(E(|cover_t(\alpha \cup \{(u_i, t))|) - E(|cover_t(\alpha)|))$ among all unselected user-cycle pairs. Then the algorithm adds (u_i, c_j) into α until the budget runs out, and can achieve the *near-optimal* solution. Suppose α_g is the solution searched by the incremental greedy algorithm [11]; then for any budget-feasible task assignment $\forall\alpha_* \subseteq R \times T$ and $b_a{*}u(\alpha_*) + b_o{*}|\alpha_*| \leq B$, we can ensure that $E(\Sigma_{t\in T}|cover_t(\alpha_g)|) \geq (1 - e^{-1})E(\Sigma_{t\in T}|cover_t(\alpha_*)|)$. For detailed proof, please refer to [11].

When $b_a > 0$ and $b_o = 0$: we can first assume that, to enjoy the free sensing cycles ($b_o = 0$), every selected participant's sensing cycles should be assigned a PCS task. Then, again, the problem becomes a submodular set function maximization over a knapsack constraint problem. A similar incremental greedy algorithm [11] selects a new participant U_i having the maximal spatial-temporal coverage increment over all sensing cycles, that is, $E(\Sigma_{t\in T}|cover_t(\alpha \cup (u_i, t))|) - E(\Sigma_{t\in T}|cover_t(\alpha)|)$. Then the algorithm adds $\{(u_i, t_*) : \forall t_* \in T\}$ into α until the budget runs out, and could be used to achieve the near-optimal solution with the same $1 - e^{-1}$ bound.

When $b_a > 0$ and $b_o > 0$: the problem becomes a submodular set function maximization over a submodular knapsack constraint problem; in this case, a nested-loop greedy algorithm [12] can achieve the near-optimal solution using $E(\Sigma_{t\in T}|cover_t(\alpha)|)$ as the objective function. Suppose α_n is the solution sought by the nested-loop greedy algorithm [12]. Given a new budget $\gamma * B$ (where γ is closed to 1), for any budget-feasible task assignment $\forall\alpha_* \subseteq R \times T$ and $b_a{*}u(\alpha^*) + b_o{*}|\alpha^*| \leq \gamma * B$, we can ensure that $E(\Sigma_{t\in T}|cover_t(\alpha_n|) \geq (1 - e^{-1})E(\Sigma_{t\in T}|cover_t(\alpha^*)|)$. For details, please refer to [11, 12]

For *Goal.2*, we can also use the greedy-based algorithms [11, 12] to solve the problem under the three corresponding b_a/b_o settings. Compared to 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 α, until $P(|cover_t(\alpha)| \geq G) \geq thr$, where $0 < thr < 1$ is a predefined threshold to bound the spatial-temporal coverage. According to the Markov inequality, we have:

$$P\left(\left|cover_t(\alpha)\right| \geq G\right) \leq \frac{E\left(\left|cover_t(\alpha)\right|\right)}{G}. \qquad (3)$$

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 we 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 Quart-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.

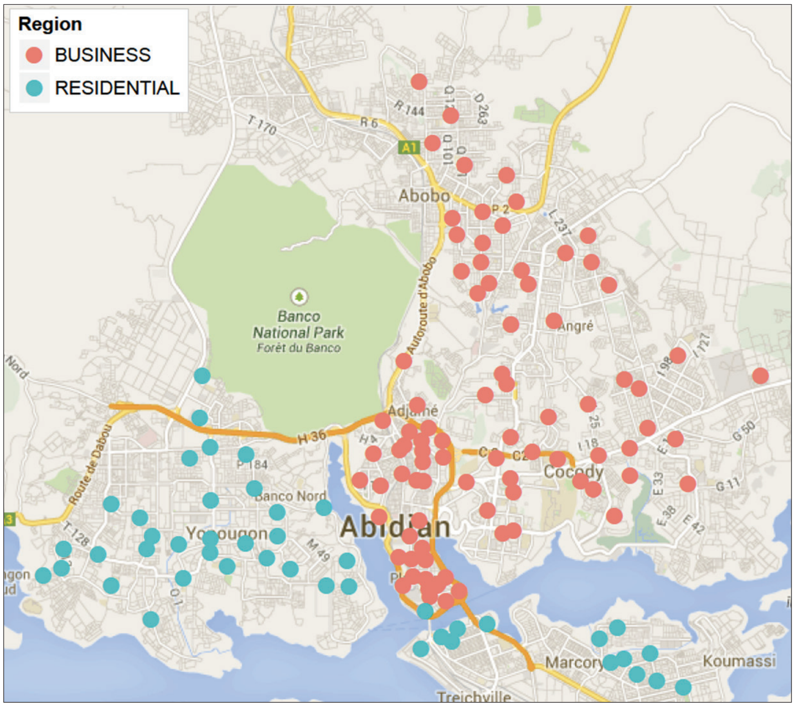

Figure 2. Testbed for the evaluation.

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 buget 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,

$$\frac{\sum_{t \in T}\left|cover_t(\alpha)\right|}{|T| * |C|} \leq 100\%.$$

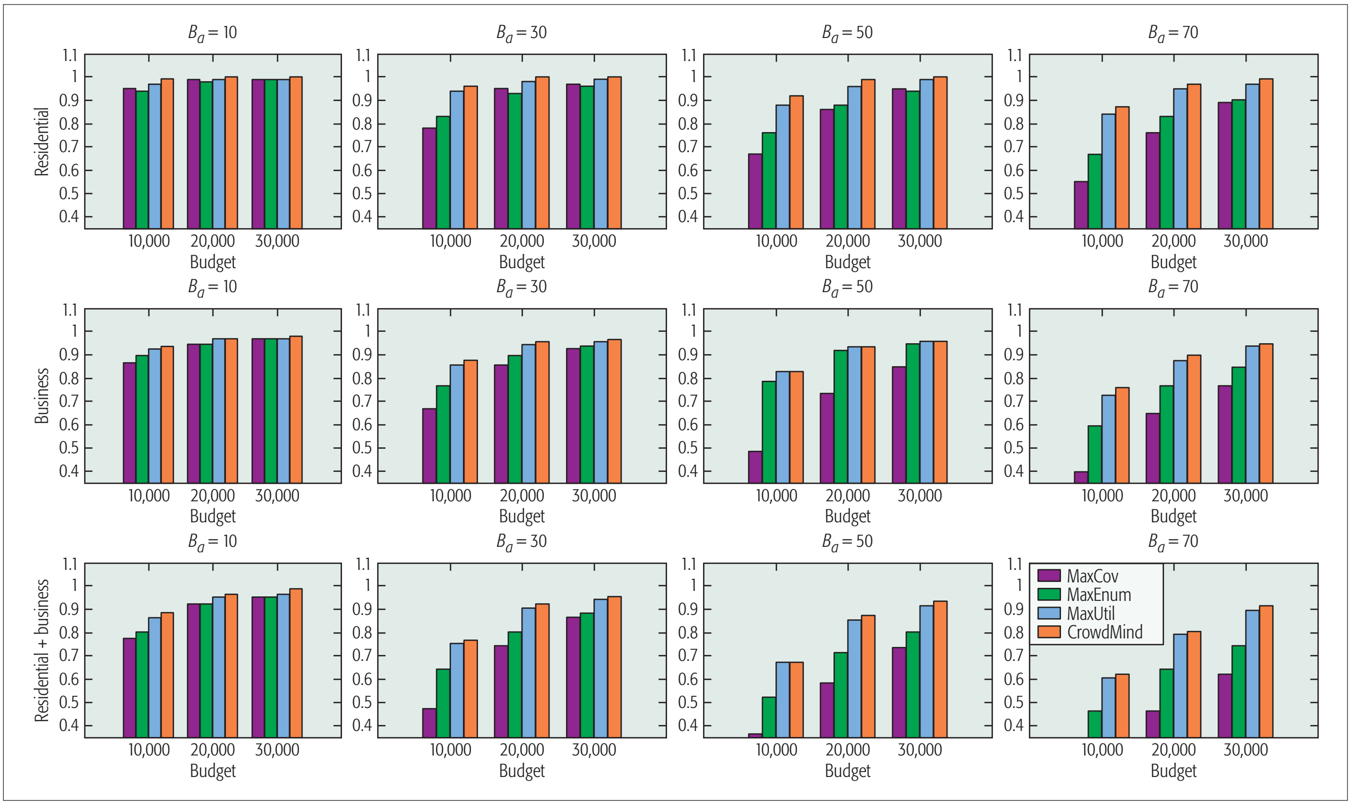

Figure 3. Evaluation results based on the three regions for Goal.1.

CrowdMind	MaxCov	MaxEnum	MaxUtil
B = 10,000			
0.83	0.49	0.83	0.79
B = 20,000			
0.94	0.74	0.94	0.92
B = 30,000			
0.96	0.85	0.96	0.95

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

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_o) settings:

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

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

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

$$\min_{t\in T}\left\{P\left(\left|cover_t(\alpha)\right|\geq G\right)\right\}$$

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 *thr* = (99.99 percent)$^{1/(|T|*|C|)}$ 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-

(a) BUSINESS region				(b) RESIDENTIAL region				(c) RESIDENTIAL+BUSINESS region			
CM.	MaxMin	MaxCom	MaxCov	CM.	MaxMin	MaxCom	MaxCov	CM.	MaxMin	MaxCom	MaxCov
$\frac{G}{\lvert C\rvert} = 95\%$				$\frac{G}{\lvert C\rvert} = 95\%$				$\frac{G}{\lvert C\rvert} = 95\%$			
537.8	601.3	951	1667.5	505.3	622.8	756	1910	776.3	862.3	1291	2442.3
$\frac{G}{\lvert C\rvert} = 85\%$				$\frac{G}{\lvert C\rvert} = 85\%$				$\frac{G}{\lvert C\rvert} = 85\%$			
258.5	293	422	564	257.3	327.8	337	617.8	321.8	377.3	557.8	997.3

Table 2. Average incentive payment (b_o = 0 and b_a = 1, where "CM." refers to CrowdMind).

mally 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 (dqzhang@sei.pku.edu.cn) 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

[1] R. K. Ganti, F. Ye, and H. Lei, "Mobile Crowdsensing: Current state and Future Challenges," *IEEE Commun. Mag.*, vol. 49, 2011, pp. 32–39.
[2] D. Zhang *et al.*, "4w1h in Mobile Crowd Sensing," *IEEE Commun. Mag.*, vol. 52, no. 8, Aug. 2014, pp. 42–48.
[3] Y. Zheng, F. Liu, and H.-P. Hsieh, "U-air: When urban Air Quality Inference Meets Big Data," *Proc. 19th ACM SIGKDD Int'l. Conf. Knowledge Discovery and Data Mining*, 2013, pp. 1436–44.
[4] X. Sheng, J. Tang, and W. Zhang, "Energy-Efficient Collaborative Sensing with Mobile Phones," *Proc. 2012 IEEE Conf. Computer Commun.*, 2012.
[5] S. Hachem, A. Pathak, and V. Issarny, "Probabilistic Registration for Large-Scale Mobile Participatory Sensing," *Proc. 2013 IEEE Int'l. Conf. Pervasive Computing and Commun.*, vol. 18, 2013, page 22.
[6] M. Musolesi *et al.*, "Supporting Energy-Efficient Uploading Strategies for Continuous Sensing Applications on Mobile Phones," *Pervasive Computing*, 2010, pp. 355–72.
[7] N. D Lane *et al.*, "Piggyback Crowdsensing (PCS): Energy Efficient Crowdsourcing of Mobile Sensor Data by Exploiting Smartphone App Opportunities," *Proc. 11th ACM Conf. Embedded Networked Sensor Systems*, 2013, page 7.
[8] D. Zhang et *al.*, "Crowdrecruiter: Selecting Participants for Piggyback Crowdsensing Under Probabilistic Coverage Constraint," *Proc. 2014 ACM Int'l. Joint Conf. Pervasive and Ubiquitous Computing*, 2014, pp. 703–14.
[9] H. Xiong *et al.*, "icrowd: Near-Optimal Task Allocation for Piggyback Crowdsensing," *IEEE Trans. Mobile Computing*, vol. 15, no. 8, 2016, pp. 2010–2022.
[10] C. W Gardiner *et al.*, "Handbook of Stochastic Methods," vol. 3, Springer Berlin, 1985.
[11] P. R. Goundan and A. S. Schulz, "Revisiting the Greedy Approach to Submodular Set Function Maximization," *Optimization Online*, 2007, pp. 1–25.
[12] R. K. Iyer and J. A. Bilmes. "Submodular Optimization with Submodular Cover and Submodular Knapsack Constraints," *Advances in Neural Information Processing Systems*, 2013, pp. 2436–44.
[13] V. D. Blondel *et al.*, "Data for Development: the d4d Challenge on Mobile Phone Data," CoRR, abs/1210.0137, 2012.
[14] M. Sviridenko "A Note on Maximizing a Submodular Set Function Subject to a Knapsack Constraint," *Operations Research Letters*, vol. 32, no. 1, 2004, pp. 41–43.
[15] S. Reddy, D. Estrin, and M. Srivastava, "Recruitment Framework for Participatory Sensing Data Collections," *Proc. Pervasive*, 2010, pages 138–55.

Biographies

Haoyi Xiong is an assistant professor in the Department of Computer Science at Missouri University of Science and Technology, Rolla. He received his Ph.D. in computer science from Télécom SudParis and Université Pierre et Marie Curie — Paris VI. His research interests include the applications of probabilistic modeling and optimization to ubiquitous computing. He received the Best Paper Award from IEEE UIC 2012 and Outstanding Ph.D. Thesis Runner-up from CNRS SAMOVAR Lab.

Daqing Zhang (dqzhang@sei.pku.edu.cn) (dqzhang@sei.pku.edu.cn) is a chair professor with the Key Laboratory of High Confidence Software Technologies, Peking University, China.. He obtained his Ph.D from the University of Rome "La Sapienza," Italy, in 1996. His research interests include context-aware computing, urban computing, mobile computing, and so on. He has served as the General or Program Chair for more than 10 international conferences. He is an Associate Editor for *ACM Transactions on Intelligent Systems and Technology*, *IEEE Transactions on Big Data*, and others.

Zhishan Guo is an assistant professor in the Department of Computer Science at Missouri University of Science and Technology. He received 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.

Guanling 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 Mobiquitous 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 (lb3dp@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.