Location Privacy-Preserving Task Allocation for Mobile Crowdsensing with Differential Geo-Obfuscation

Leye Wang
Hong Kong University of Science and Technology
wly@cse.ust.hk

Dingqi Yang
University of Fribourg
dingqi.yang@unifr.ch

Xiao Han
Shanghai University of Finance and Economics
xiaohan@mail.shufe.edu.cn

Tianben Wang
Northwestern Polytechnical University
wangtianbengx@163.com

Daqing Zhang
Key Lab of High Confidence Software Technologies, Peking University
Telecom SudParis
dqzhang@sei.pku.edu.cn

Xiaojuan Ma
Hong Kong University of Science and Technology
mxj@cse.ust.hk

ABSTRACT

In traditional mobile crowdsensing applications, organizers need participants' precise locations for optimal task allocation, e.g., minimizing selected workers' travel distance to task locations. However, the exposure of their locations raises privacy concerns. Especially for those who are not eventually selected for any task, their location privacy is sacrificed in vain. Hence, in this paper, we propose a location privacy-preserving task allocation framework with geo-obfuscation to protect users' locations during task assignments. Specifically, we make participants obfuscate their reported locations under the guarantee of differential privacy, which can provide privacy protection regardless of adversaries' prior knowledge and without the involvement of any third-part entity. In order to achieve optimal task allocation with such differential geo-obfuscation, we formulate a mixed-integer non-linear programming problem to minimize the expected travel distance of the selected workers under the constraint of differential privacy. Evaluation results on both simulation and real-world user mobility traces show the effectiveness of our proposed framework. Particularly, our framework outperforms Laplace obfuscation, a state-of-the-art differential geo-obfuscation mechanism, by achieving 45% less average travel distance on the real-world data.

Keywords

Crowdsensing; task allocation; differential location privacy

1. INTRODUCTION

Mobile crowdsensing (MCS) is a promising sensing paradigm which leverages the power of rich-sensor-equipped smartphones [13, 34]. It has attracted the interests of both academia and industry, and enables various real-world applications, such as environment monitoring [24] and point-of-interest characterization [9]. As a representative, the commercial crowdsensing-based traffic monitoring and route navigation app, *WAZE*, has already achieved more than 100 million downloads with a user score up to 4.6/5 on Google Play [2].

On a typical MCS platform, users are registered as candidate workers. When MCS new tasks come, the platform selects a proper subset of candidates to complete the tasks by paying them some incentives. This worker selection process, called *task allocation*, is a key step in MCS that can significantly impact the efficiency of MCS. Particularly, workers' *travel distance* to task locations is an important issue to consider in task allocation. For participants, if the travel distance is too long, they will probably be unwilling to conduct the task. For task organizers, long travel distance will lead to unsatisfactory conditions such as high incentive to pay and large sensing delay. Therefore, in this paper, following previous work [15, 26], we use travel distance as the utility metric for task allocation.

Existing work on MCS task allocation mostly assumes that candidates' locations are known to the platform, and thus can optimize the task efficiency (i.e., minimize the travel distance) by directly assigning tasks to nearby workers. However, this also indicates that users' *location privacy* is at risk. Note that in task allocation, only a subset of candidates are selected as workers while all of them are requested to share their locations. Even though the selected workers' privacy concerns may be alleviated with incentives, there is no compensation for the privacy sacrifice of the remaining candidates. These people may get discouraged and leave the MCS platform, downsizing the candidate worker pool and impairing the performance of the whole platform. Therefore, location privacy needs to be carefully considered in task allocation, especially for the large number of unselected candidates.

While researchers begin to address the interdisciplinary topic of optimizing MCS task allocation under location privacy protection in recent years, most existing solutions still suffer from the following limitations.

(1) **Sensitive to adversaries' prior knowledge**. According to a recent survey [22], most existing mechanisms (e.g. [10, 23, 27]) employ a cloaking-based idea (i.e., using

WWW 2017, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4913-0/17/04.
http://dx.doi.org/10.1145/3038912.3052696

a coarse area to represent a precise location) to provide location protection, but their expected privacy guarantee can be easily downgraded if adversaries hold certain prior knowledge [3]. For example, if an adversary foreknows that a user is a student, and the cloaking area includes both a school and government office, the adversary can confidently infer that the user is in the school region.

(2) **Dependent on third-party trusted entities**. Some existing mechanisms require the support of other third-parties (besides users and MCS platforms), which makes them difficult to deploy in reality. For example, To et. al [26] need users' cellular service providers to take an important coordination role between users and MCS platforms to provide privacy protection, while in practice cellular service providers may lack motivation to participate.

Therefore, MCS is still in need for a more competitive and practical location privacy-preserving task allocation mechanism, which can robustly protect users' privacy against adversaries holding arbitrary prior knowledge without involving third-parties.

Recently, location privacy research introduces *differential privacy* [12] to provide theoretically guaranteed protection regardless of adversaries' prior knowledge. Consequently, some Location-Based Services (LBS) have proposed several *differential geo-obfuscation* mechanisms [3, 7]. Such approaches in LBS shed lights on the design of privacy-preserving MCS task allocation regarding the two aforementioned concerns. First, differential privacy ensures that the probability of users being mapped to one specific obfuscated location from any of the actual locations is similar, so that an adversary with any prior knowledge gains little additional information from the observation (i.e., obfuscated location). Second, differential geo-obfuscation alters users' locations on their smartphones, and thus has no need to involve trustful third-parties.

However, compared to LBS, optimizing MCS task allocation under differential geo-obfuscation needs to address a new challenge. Specifically, in contrast to LBS where each individual user's geo-obfuscation method can be optimized independently by considering only his/her own actual and obfuscated locations [7], the utility of MCS task allocation depends on all the candidates' locations, and thus the optimization process must collectively take all the candidates into account. For example, suppose there are two candidates u_1, u_2 and one task t_1, and u_1 is the one nearer to the location of t_1 (should be selected as worker). After geo-obfuscation, task allocation may wrongly select u_2 as the worker if u_1's perturbed location is farther away from t_1's location than that of u_2. With this in mind, both u_1 and u_2's (obfuscated) locations, as well as t_1's location, need to be considered in designing the task allocation mechanism and geo-obfuscation function. In reality, as many candidates and tasks will simultaneously co-exist, it is rather challenging to optimally incorporate differential geo-obfuscation in MCS task allocation while minimizing the workers' overall travel distance.

In this paper, we propose an MCS task allocation framework to protect participants' location privacy with differential geo-obfuscation, while minimizing the selected workers' overall travel distance. The contributions of this paper can be summarized as follows.

(1) To the best of our knowledge, this is the first work to introduce differential geo-obfuscation to MCS task allocation, so as to protect participants' location privacy regardless of adversaries' prior knowledge and avoid involving any third-party in the process.

(2) To minimize the travel distance under privacy protection, we propose an optimal privacy-preserving MCS task allocation framework with two interleaved modules: *differential geo-obfuscation* and *obfuscation-aware task allocation*. We formulate a *mixed-integer nonlinear program (MINLP)* to collectively optimize the two modules by minimizing the expected travel distance of the selected workers while ensuring the completion of all tasks available. As directly solving MINLP is NP-hard, we use the *Benders Decomposition* [5] method to decompose it into two *linear programs (LP)*, each of which corresponds to optimizing one module while fixing the other. We then iteratively optimize the two LPs till convergence to obtain the final solution.

(3) The evaluation on both simulation and real-world user mobility traces verifies that our proposed privacy-preserving framework can reduce up to 45% average travel distance of selected workers compared to a state-of-the-art differential geo-obfuscation mechanism, Laplace obfuscation [3].

2. BACKGROUND

In this section, we first clarify the MCS task model studied in this paper, followed by the basic concepts of differential geo-obfuscation.

2.1 Mobile Crowdsensing Task Model

In MCS, there are two task assignment models [26], *Worker Selected Task* (WST) and *Server Assigned Task* (SAT). In WST model, the MCS platform publishes tasks online and candidates can select any tasks to conduct without exposing their location information. In SAT model, candidates upload their locations to the platform and the platform selects some candidates to allocate tasks. Although WST model is more friendly to users' privacy, it falls shorts in not being able to globally control the task allocation process. In contrast, SAT can better optimize the overall efficiency of all the MCS tasks as the platform has the overall knowledge of all the candidates' locations. This paper attempts to combine the advantages of both models, i.e., using SAT to get good running performance of all the MCS tasks, while still protecting users' location privacy.

Moreover, in this paper, we assume that the number of tasks is smaller than that of candidates on the MCS platform, so no selected worker needs to perform more than one task in one snapshot of the task allocation. This assumption is reasonable as today's milestone MCS applications have already attracted millions of users (e.g. WAZE [2]), and limiting the number of tasks for each user can benefit both the quality of task performing and user fairness [25].

2.2 Differential Geo-Obfuscation

Differential privacy is recently introduced in location protection by Andres et. al [3]. It performs as a probabilistic geo-obfuscation process, i.e., a user first obfuscates his real location to another one according to a pre-configured probability function P (encoding the probability of mapping arbitrary location l to l^*) and then uploads the obfuscated location to the server. The probability function is the key to ensure differential privacy. The basic idea is that, suppose the obfuscated location is l^*, for any two locations l_1, l_2, their probability of being mapped to l^* are **similar**. Then,

if an adversary observes a user u in l^*, he/she cannot distinguish whether u is actually in l_1 or l_2, even if he/she knows the obfuscation function P. With this intuition, differential privacy formally defines such **similarity** between *any* two locations l_1, l_2 for *arbitrary* l^*.

Differential Privacy [3, 7]. *Suppose the concerned area includes a set of locations $\mathcal{L}$, then a probabilistic geo-obfuscation function P satisfies ϵ-differential-privacy, iff.*

$$P(l^*|l_1) \leq e^{\epsilon d(l_1,l_2)} P(l^*|l_2) \qquad \forall l_1, l_2, l^* \in \mathcal{L} \qquad (1)$$

where $P(l^*|l)$ is the probability of obfuscating l to l^*, $d(l_1, l_2)$ is the distance between l_1 and l_2, ϵ is the privacy budget — the smaller ϵ, the higher privacy.

The distance $d(l_1, l_2)$ is introduced in the formulation to reflect the intuition that if l_1 and l_2 are close to each other (i.e., small $d(l_1, l_2)$), they should be more indistinguishable. Note that the set of locations $\mathcal{L}$ can be constructed by dividing the concerned area into a set of regions (of arbitrary size) and selecting the representative locations of the regions (e.g., geographic center) [7]. While $d(l_1, l_2)$ could be any distance metric theoretically, following [7], we consider $d(l_1, l_2)$ as Euclidean distance with the unit of kilometer.

If P satisfies ϵ-differential-privacy, it can be theoretically proved that with the observation of the obfuscated location l^*, the improvement of an adversary's posterior knowledge about a user's location distribution σ over the prior distribution π, i.e., σ/π, is bounded by $e^{\epsilon D(\mathcal{L})}$ ($D(\mathcal{L})$ is the maximum distance of any two locations in $\mathcal{L}$), regardless of what the prior π is [3]. Thus, differential geo-obfuscation can robustly protect users' location privacy against adversaries with arbitrary prior knowledge. Please refer to [3] for the theoretical proof.

3. PROBLEM ANALYSIS

In this section, we first illustrate the overall process of MCS task allocation with differential geo-obfuscation. Then, we formalize the key problems during this process.

3.1 Task Allocation with Geo-Obfuscation

Suppose there exists an MCS platform holding various sensing tasks (e.g., noise and air quality sensing) in a certain city which require workers to conduct. To protect users' privacy, rather than frequently requiring location updating, our framework only needs candidates to upload their (obfuscated) locations before a snapshot of task allocation, which is called *initialization stage* (e,g., 1-hour snapshot with 5-minute initialization). More specifically, the initialization stage first generates a geo-obfuscation function (considering task locations), and transfers this function to candidates, and then collects their obfuscated locations. The non-responding candidates can be seen as unavailable, so that this initialization stage is also an effective step to filter out unavailable candidates. Finally, after collecting available candidates' obfuscated locations, we assign tasks to appropriate candidates.

Briefly, the above running process includes three steps, as shown in Figure 1: (1) *Platform-side Geo-Obfuscation Function Generation*, (2) *User-side Location Obfuscation*, and (3) *Platform-side Obfuscation-aware Task Allocation*.

(1) *Platform-side Geo-Obfuscation Function Generation.* Before collecting candidates' locations, a probabilistic obfuscation function needs to be generated for candidates with

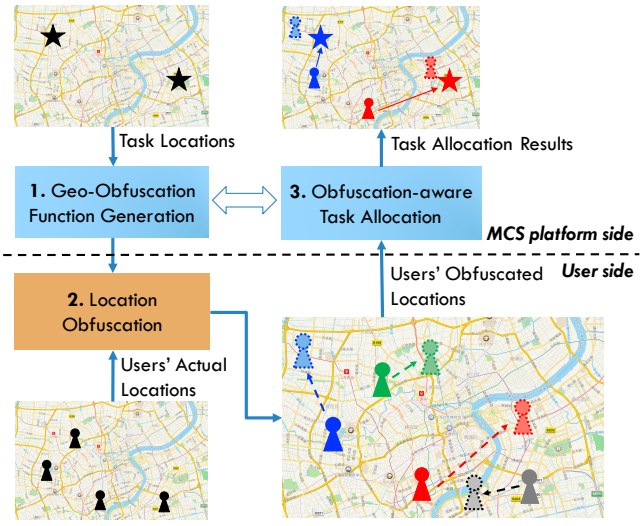


Figure 1: Workflow of task allocation with geo-obfuscation.

certain differential privacy guarantee. Note that task locations need to be considered when generating the geo-obfuscation function, as we attempt to reduce the negative effects of such geo-obfuscation on the workers' travel distance to task locations. Besides, the platform can take charge of generating the obfuscation function without violating users' privacy, since the theoretical protection of differential privacy is guaranteed assuming that the adversary knows the obfuscation function [3]. In other words, the platform knows no more than an adversary, so that users can get privacy protection without needing to trust the platform (even it generates the obfuscation function).

(2) *User-side Location Obfuscation.* After the platform generates the obfuscation function, the candidates can download it into their smartphones, and then obfuscate their actual locations according to the probabilities encoded in the function. The obfuscated locations are uploaded to the platform for task allocation in the next step. Since the location obfuscation runs completely locally in a user's smartphone, no one else knows the user's real location.

(3) *Platform-side Obfuscation-aware Task Allocation.* Finally, after receiving candidates' obfuscated locations, the MCS platform will assign tasks to proper workers, attempting to minimize the selected workers' travel distance to the task locations. Since users' uploaded locations are obfuscated, directly seeing them as actual locations and allocating tasks may not perform well. Instead, the obfuscation function may be taken into account for better task allocation efficiency.

Note that to minimize workers' travel distance, the design of geo-obfuscation function and task allocation are somehow interleaved. In other words, the task allocation could be optimized only when the geo-obfuscation function is given, and vice-versa. Therefore, collectively optimizing these two parts is necessary to ensure a good system utility. Next, we will mathematically formalize these two key problems.

3.2 Mathematical Problem Formulation

In this section, we formally define the two key problems in the above process: *differential geo-obfuscation* and *obfuscation-aware task allocation.*

3.2.1 Differential Geo-Obfuscation Function

Briefly, the problem of generating the geo-obfuscation function P can be formulated as:

minimize: *Travel distance of selected workers*
subject to: *P satisfies differential privacy*

As the differential privacy constraint has been given in (1), we still need to mathematically model the travel distance of all the selected workers. To this end, we first calculate the expected travel distance of assigning a task at l_t to a user at (obfuscated) l^* given the geo-obfuscation function P.

$$d^*(l^*, l_t) = \frac{\sum_{l\in\mathcal{L}} \pi(l)P(l^*|l)d(l, l_t)}{\sum_{l\in\mathcal{L}} \pi(l)P(l^*|l)} \tag{2}$$

where π is the candidates' overall geographic distribution in the concerned set of locations $\mathcal{L}$ ($\sum_{l\in\mathcal{L}} \pi(l) = 1$), and how to estimate it will be elaborated in Sec. 4; $d(l, l')$ is the distance between locations l and l'.

Suppose $x(l^*, l_t)$ denotes the number of task assignments which allocate the tasks at l_t to the users at l^*. Based on x, we can calculate the sum of expected travel distances of all the selected users as:

$$\sum_{l^*\in\mathcal{L}} \sum_{l_t\in\mathcal{L}} d^*(l^*, l_t)x(l^*, l_t) \tag{3}$$

$$= \sum_{l^*\in\mathcal{L}} \sum_{l_t\in\mathcal{L}} \frac{\sum_{l\in\mathcal{L}} \pi(l)P(l^*|l)d(l, l_t)}{\sum_{l\in\mathcal{L}} \pi(l)P(l^*|l)} x(l^*, l_t) \tag{4}$$

Note that when we optimize the geo-obfuscation function P, the actual task allocation result x is unknown. This means that P has to be optimized under a certain *hypothetical* x. More specifically, to minimize (4), this hypothetical x is also a variable to be optimized, i.e., P and x are the best combination to achieve the minimal (4). We denote this x in the optimal combination $\{P, x\}$ as $\hat{x}$.

Then, given the number of tasks at each location l, denoted as $N_t(l)$, and the total number of candidates N_c[1], we can mathematically formalize the problem of optimizing geo-obfuscation function P as:

$$\min_{P,\hat{x}} \sum_{l^*\in\mathcal{L}} \sum_{l_t\in\mathcal{L}} \frac{\sum_{l\in\mathcal{L}} \pi(l)P(l^*|l)d(l, l_t)}{\sum_{l\in\mathcal{L}} \pi(l)P(l^*|l)} \hat{x}(l^*, l_t) \tag{5}$$

$$\text{s.t.} \quad P(l^*|l_1) \le e^{\epsilon d(l_1,l_2)} P(l^*|l_2) \qquad l_1, l_2, l^* \in \mathcal{L} \tag{6}$$

$$\sum_{l^*\in\mathcal{L}} \hat{x}(l^*, l_t) = N_t(l_t) \qquad l_t \in \mathcal{L} \tag{7}$$

$$\sum_{l\in\mathcal{L}} \pi(l)P(l^*|l) = \pi(l^*) \qquad l^* \in \mathcal{L} \tag{8}$$

$$\sum_{l_t\in\mathcal{L}} \hat{x}(l^*, l_t) \le \pi(l^*)N_c \qquad l^* \in \mathcal{L} \tag{9}$$

$$\sum_{l^*\in\mathcal{L}} P(l^*|l) = 1 \qquad l \in \mathcal{L} \tag{10}$$

$$P(l^*|l) \ge 0 \qquad l, l^* \in \mathcal{L} \tag{11}$$

$$\hat{x}(l^*, l_t) \in \mathbb{Z}_{\ge 0} \qquad l^*, l_t \in \mathcal{L} \tag{12}$$

As above mentioned, although we attempt to optimize the geo-obfuscation function (P), the hypothetical task allocation scheme ($\hat{x}$) also needs to be optimized. Eq. (6) is the constraint of differential privacy; Eq. (7) guarantees that every task has a worker; Eq. (8) ensures that the geo-obfuscation does not change candidates' overall location distribution[2]; according to the task model discussed in Sec. 2.1, Eq. (9) ensures that the number of tasks assigned to users at l^* is smaller than the total number of users there, so that no worker needs to do more than one task[3]; Eq. (10) and (11) are two general probability constraints; Eq. (12) ensures that the number of task allocations should be integer. As the constraints include integral restrictions (12) and the objective function (5) is non-linear with respect to the variables P and $\hat{x}$, this optimization is a *mixed-integer non-linear program* (MINLP) [4]. While state-of-the-art non-linear optimization techniques can deal with *convex* objectives effectively [8], unfortunately, our objective function is non-convex. To this end, a specialized algorithm is required to solve this MINLP for getting a (near) optimal geo-obfuscation function, which will be presented in Sec. 4.

3.2.2 Obfuscation-aware Task Allocation

The above formulation is used for generating the obfuscation function (although it is constructed based on the hypothetical optimal task allocation). After the candidates upload their obfuscated locations, the server needs to actually allocate tasks according to the users' uploaded locations. We denote such real task allocation scheme as $\tilde{x}$, and the problem of optimizing $\tilde{x}$ is formalized as:

$$\min_{\tilde{x}} \sum_{l^*\in\mathcal{L}} \sum_{l_t\in\mathcal{L}} \frac{\sum_{l\in\mathcal{L}} \pi(l)P(l^*|l)d(l, l_t)}{\sum_{l\in\mathcal{L}} \pi(l)P(l^*|l)} \tilde{x}(l^*, l_t) \tag{13}$$

$$\text{s.t.} \quad \sum_{l^*\in\mathcal{L}} \tilde{x}(l^*, l_t) = N_t(l_t) \qquad l_t \in \mathcal{L} \tag{14}$$

$$\sum_{l_t\in\mathcal{L}} \tilde{x}(l^*, l_t) \le N_c(l^*) \qquad l^* \in \mathcal{L} \tag{15}$$

$$\tilde{x}(l^*, l_t) \in \mathbb{Z}_{\ge 0} \qquad l^*, l_t \in \mathcal{L} \tag{16}$$

where $N_c(l^*)$ is the actual number of users with obfuscated location l^*. The objective is still minimizing the travel distance, while P is already known and the only variable is $\tilde{x}$. Hence, this is a *mixed-integer linear program* (MILP). Solving MILP is much easier than MINLP, and many up-to-date optimization tools can solve it efficiently with well-studied optimization techniques such as *branch and bound* [19]. Based on $\tilde{x}(l^*, l_t)$, which points out how many candidates at obfuscated l^* will be selected to conduct the task at l_t, we can then randomly select this number of workers from all the candidates reporting their locations as l^*.

4. GEO-OBFUSCATION OPTIMIZATION

As analyzed in the previous section, the first step of our framework, i.e., geo-obfuscation function optimization, needs a specialized algorithm for solving the relevant MINLP (5).

[1]We can get N_c by sending a message to all the users on the platform and collects their feedbacks before generating the geo-obfuscation function.

[2]Keeping important statistics invariant in obfuscation is a common practice in statistical disclosure control with many benefits [31]. In our case, for instance, this ensures that directly plotting candidates' obfuscated locations on the map can still roughly reflect the user distribution. Such a map is usually an important part of the user interface for MCS applications (e.g., WAZE).

[3]Because we cannot foreknow the real number of users whose obfuscated location is l^*, here we can just estimate it using the overall geo-distribution and total number of candidates.

Our overall strategy for optimizing geo-obfuscation function is to decompose the original MINLP (5) into two linear programs (LP). Each LP corresponds to optimizing one of the variables P or x while fixing the other. Then, we can iteratively solve the two LPs until convergence to get a resultant P. This strategy is widely known as *Benders Decomposition* (BD) [5]. While BD solution depends on the initial value of P or $\hat{x}$, the optimized result may fall in a *local optima*. To relieve this pitfall, we adopt a *Genetic Algorithm* (GA) to progressively choose better initial values of P or $\hat{x}$ to yield shorter travel distances.

4.1 Benders Decomposition

BD is an optimization technique first proposed for solving very large scale linear programming problems [5], and later is extended to solve mixed-integer nonlinear programming problems [14]. The basic idea is *divide-and-conquer*, i.e., dividing the variables into two subsets so that two subproblems are derived. Then, the solution of one subproblem can be seen as the input of another subproblem, and the two subproblems are alternatively solved until convergence (or the iteration times exceed a given threshold).

As our geo-obfuscation optimization intrinsically includes two subsets of variables, P and $\hat{x}$, we can then accordingly split the original optimization problem into two subproblems of solving P and $\hat{x}$, respectively. Each subproblem only includes the constraints relevant to either P or $\hat{x}$.

P-subproblem:

$$\min_{P} \sum_{l^* \in \mathcal{L}} \sum_{l_t \in \mathcal{L}} \frac{\sum_{l \in \mathcal{L}} \pi(l) P(l^*|l) d(l, l_t)}{\sum_{l \in \mathcal{L}} \pi(l) P(l^*|l)} \hat{x}(l^*, l_t) \tag{17}$$

$$\text{s.t.} \quad P(l^*|l_1) \leq e^{\epsilon d(l_1, l_2)} P(l^*|l_2) \qquad l_1, l_2, l^* \in \mathcal{L} \tag{18}$$

$$\sum_{l \in \mathcal{L}} \pi(l) P(l^*|l) = \pi(l^*) \qquad l^* \in \mathcal{L} \tag{19}$$

$$\sum_{l^* \in \mathcal{L}} P(l^*|l) = 1 \qquad l \in \mathcal{L} \tag{20}$$

$$P(l^*|l) \geq 0 \qquad l, l^* \in \mathcal{L} \tag{21}$$

Note that the objective (17) can be converted as follows, by considering (19):

$$\min_{P} \sum_{l^* \in \mathcal{L}} \sum_{l_t \in \mathcal{L}} \sum_{l \in \mathcal{L}} \frac{\pi(l)}{\pi(l^*)} d(l, l_t) \hat{x}(l^*, l_t) P(l^*|l) \tag{22}$$

Given $\hat{x}$, the objective (22) is a linear function regarding P, and thus P-subproblem is a linear programming problem.

$\hat{\mathbf{x}}$-subproblem:

$$\min_{\hat{x}} \sum_{l^* \in \mathcal{L}} \sum_{l_t \in \mathcal{L}} \frac{\sum_{l \in \mathcal{L}} \pi(l) P(l^*|l) d(l, l_t)}{\sum_{l \in \mathcal{L}} \pi(l) P(l^*|l)} \hat{x}(l^*, l_t) \tag{23}$$

$$\text{s.t.} \quad \sum_{l^* \in \mathcal{L}} \hat{x}(l^*, l_t) = N_t(l_t) \qquad l_t \in \mathcal{L} \tag{24}$$

$$\sum_{l_t \in \mathcal{L}} \hat{x}(l^*, l_t) \leq \pi(l^*) N_c \qquad l^* \in \mathcal{L} \tag{25}$$

$$\hat{x}(l^*, l_t) \in \mathbb{Z}_{\geq 0} \qquad l^*, l_t \in \mathcal{L} \tag{26}$$

Given P, the objective (23) is a linear function regarding $\hat{x}$; considering the integral constraint (26), $\hat{x}$-subproblem is then a mixed-integer linear programming problem[4].

[4] $\hat{x}$-subproblem is similar to the task allocation problem (Sec. 3.2.2) except for the difference between (15) and (25), as we do not know real user number in each obfuscated region when solving $\hat{x}$-subproblem.

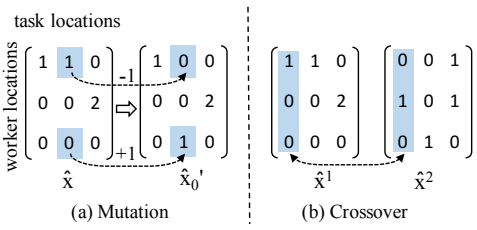


Figure 2: Illustrative examples of mutation and crossover

In a word, after the Benders Decomposition, P-subproblem and $\hat{x}$-subproblem are both changed to (mixed-integer) linear programming problems, which can be efficiently solved with off-the-shelf linear optimization softwares. In our experiment, we find that usually after three or four iterations, the iterative problem solving process is converged.

4.2 Genetic Algorithm based Initialization

To start the iteration of solving P-subproblem and $\hat{x}$-subproblem, we need to set an initial $\hat{x}$ (if solving P-subproblem first) or P (if solving $\hat{x}$-subproblem first), denoted as $\hat{x}_0$ or P_0. As using BD to optimize the geo-obfuscation function often leads to the local optima, the selection of the initial value of $\hat{x}_0$ or P_0 becomes important regarding how good the local optima can achieve.

To address this issue, we adopt a Genetic Algorithm (GA) [21] to select the initial values of $\hat{x}_0$ that deserve testing based on the previously obtained local optima $\hat{x}$.[5] Based on the new $\hat{x}_0$, we can learn P, and followed by the iterative BD process for geo-obfuscation optimization. The key idea of GA is to generate a potential solution for utility testing from existing solutions by using either *mutation* or *crossover* methods under a given probability, which is often set according to specific applications [21]. We design the mutation and crossover processes as follows (examples in Figure 2).

Mutation: Given a previous obtained $\hat{x}$, we randomly select a location pair $(l_1, l_2) \in \{(l, l') | \hat{x}(l, l') > 0\}$. Afterward, we randomly select another location l_3 ($l_3 \neq l_1$). Then we construct a new $\hat{x}_0'$ by setting $\hat{x}_0'(l_1, l_2) = \hat{x}(l_1, l_2) - 1$, $\hat{x}_0'(l_3, l_2) = \hat{x}(l_3, l_2) + 1$, and the rest values same as $\hat{x}$.

Crossover: Given the parents $\hat{x}^1$ and $\hat{x}^2$, the crossover function is used to generate two children $\hat{x}_0^{1'}$ and $\hat{x}_0^{2'}$ by column exchange. More specifically, we randomly select a location l' and then set $\hat{x}_0^{1'}(:, l') = \hat{x}^2(:, l')$ and $\hat{x}_0^{2'}(:, l') = \hat{x}^1(:, l')$; for the rest values, $\hat{x}_0^{1'} = \hat{x}^1$ and $\hat{x}_0^{2'} = \hat{x}^2$.

Note that for both mutation and crossover results, the constraint (25) may be violated, i.e., the number of selected workers may be larger than the number of candidates in a certain location. Therefore, we need to recheck whether (25) stands after mutation or crossover. If not standing, we will re-run mutation or crossover until (i) the constraint (25) stands, or (ii) the re-run times exceed a given threshold.

4.3 Candidate Geo-Distribution Estimation

Our optimization process needs the overall geographic distribution of candidates, π, as one input. In reality, the exact

[5] Using GA to construct a new feasible P is complicated due to the existence of differential privacy constraint (18). We thus focus on generating new $\hat{x}_0$.

π is hardly known, especially as candidates upload their obfuscated locations. Here, we propose a method to estimate π based on candidates' previously uploaded obfuscated locations. In such a way, when a new round of task allocation starts, we always use an up-to-date approximation of π based on candidates' obfuscated locations in previous rounds.

In principle, a candidate's actual location l could be considered as a random sample from all the locations $\mathcal{L}$ according to π. Although his/her reported location is obfuscated, it can still help to improve our estimation about π, especially because the obfuscation function P is known to the MCS platform. Hence, estimating π can be seen as a process of gradually updating π according to candidates' new-coming reported obfuscated locations. Then, this can be modeled using Bayesian analysis. Suppose a user's obfuscated location is l^*, and the corresponding obfuscation function is P, we can update π according to the Bayes Rule as:

$$\pi(l) \leftarrow \frac{\pi(l)P(l^*|l)}{\sum_{l' \in \mathcal{L}} \pi(l')P(l^*|l')}, \quad l \in \mathcal{L} \tag{27}$$

At the beginning, we need to set an initial value to π, denoted as π_0. In most cases, π_0 can be chosen as non-informative uniform distribution, or the overall population distribution over the target sensing area (e.g., modeled by mobile phone call traces [33]). With the continuously incoming observations (i.e., obfuscated locations), the estimated π will converge to the real candidate geo-distribution, and the impact of π_0 on the estimated π is gradually reduced [17].

Note that this estimation method has an implicit assumption that candidates' actual locations are sampled from the same hidden geographic distribution. In reality, users' mobility patterns could be affected by various *contexts* [11]; only under similar contexts, this assumption could stand. Therefore, in implementation, we can estimate a set of π corresponding to various contexts (e.g. time, weekday or holiday [33]). A candidate's uploaded obfuscated location is only used to infer the π under its corresponding context.

4.4 Implementation Speedup

The ϵ-differential-privacy constraint (18) of P-subproblem involves $O(|\mathcal{L}|^3)$ constraints, which makes the optimization process hard to be extended to a large set of $\mathcal{L}$. Therefore, we adopt a δ-spanner-based approximation method to reduce the number of constraints to $O(|\mathcal{L}|^2)$, which is proposed in [7]. The basic idea is to compare only a subset of location pairs (which is specified by the edges of a δ-spanner graph) with a stricter $\frac{\epsilon}{\delta}$-differential-privacy constraint, so as to still ensure all the location pairs satisfying ϵ-differential-privacy. That is, we can replace (18) with the following constraint:

$$P(l^*|l_1) \leq e^{\frac{\epsilon}{\delta} d(l_1,l_2)} P(l^*|l_2) \qquad l^* \in \mathcal{L}, (l_1, l_2) \in \mathcal{E} \tag{28}$$

where $\mathcal{E}$ is the set of edges in the δ-spanner graph. It has been proved that for any $\delta > 1$, we can generate a δ-spanner graph with $O(\frac{|\mathcal{L}|}{\delta - 1})$ edges [7], so that the number of constraints can be reduced to $O(|\mathcal{L}|^2)$. Please refer to [7] for more details. Following [7], we set δ to 1.05 in this work.

5. EVALUATION

In this section, we assess the effectiveness of our proposed framework in two aspects. First, we evaluate the performance of our framework by simulating a target sensing area and candidates' real locations. The advantage of simulation is that we can control different key parameters (e.g., the area size and the candidate spatial distribution) and investigate how our framework performs when they vary. Second, to validate its applicability in real-world use cases, we also verify our framework on a real-life mobility dataset, D4D [6], which includes 50,000 users' two-week mobility traces represented by their mobile phone call logs.

Table 1: Key parameters in simulation.

Notation	Default	Description
n	4	side length of area
N_c	10	candidate number
N_t	4	task number
ϵ	ln(4)	differential privacy level
π	uniform	candidate spatial distribution
τ	uniform	task spatial distribution

5.1 Experiment Setup

5.1.1 Evaluation Scenarios

Simulation. We simulate a target area with $n \times n$ grids and the collection of all the grid centers forms the whole location set $\mathcal{L}$. Each grid is set to 1km*1km. We vary six key parameters in Table 1 to evaluate our framework under different settings.

D4D [6]. D4D dataset includes 50,000 users' phone call traces in Costa d'Ivori, which is widely used to evaluate task allocation mechanisms in MCS [15, 20, 35]. Referring to [15, 20], we see a user's current location as the position of the cell tower where he/she makes the last phone call. We select the downtown area of the largest city in Costa d'Ivori, *Abidjan*, as the target area, and randomly distribute tasks to a group of cell towers within the area.

5.1.2 Baselines

Laplace. We compare our framework with the state-of-the-art differential obfuscation mechanism [3] that adds Laplacian noise to a user's actual location, denoted as *Laplace*. Intuitively, Laplace tends to obfuscate a location to its nearby locations with higher probabilities. Formally, the obfuscation probabilities are:

$$P(l^*|l) \propto e^{-\epsilon \frac{d(l,l^*)}{D(\mathcal{L})}} \tag{29}$$

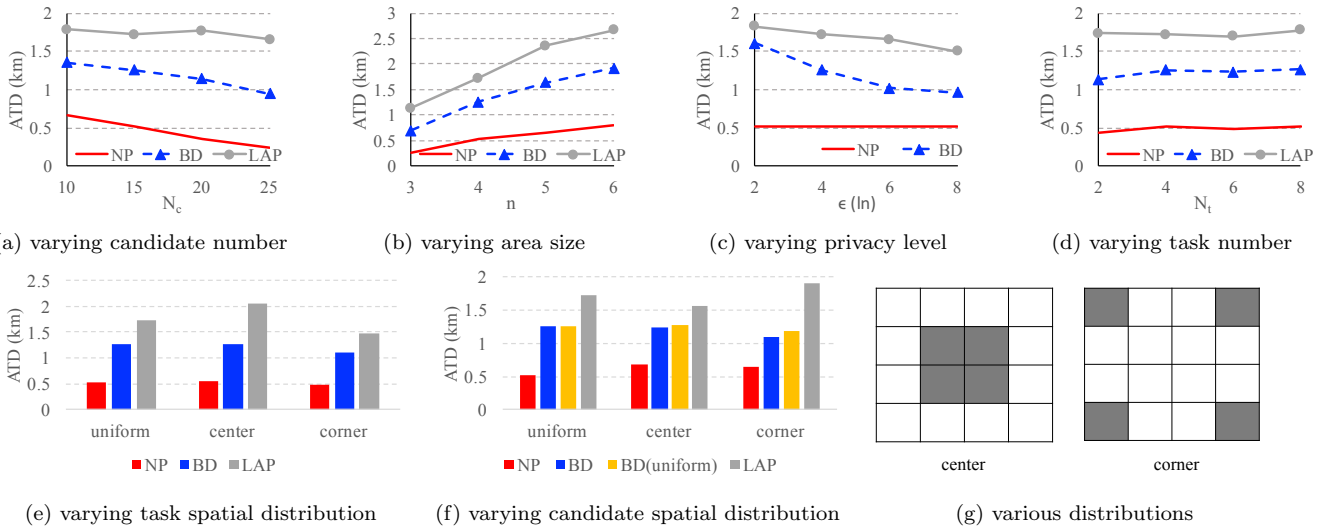
where $D(\mathcal{L})$ is the maximum distance between any two locations in the target area $\mathcal{L}$. Note that to make the comparison fair, the task allocation part of Laplace also adopts the same linear program illustrated in Sec. 3.2.2 to get the *optimal* task assignments.

No-Privacy. We also show the optimal location allocation results when candidates' real locations are reported, which can be seen as a lower bound of ATD for location privacy-preserving task allocation mechanisms.

5.1.3 Evaluation Metric

Referring to [16], we use the Euclidean distance to measure the travel distance needed for workers to complete a task. The evaluation metric for the task allocation efficiency is then the Average Travel Distance (ATD) of the selected workers and their assigned task locations:

$$ATD = \sum_{(u,t) \in \mathcal{A}} d(u,t)/|\mathcal{A}| \tag{30}$$



(a) varying candidate number (b) varying area size (c) varying privacy level (d) varying task number

(e) varying task spatial distribution (f) varying candidate spatial distribution (g) various distributions

Figure 3: Evaluation results on simulation. (NP: no-privacy, BD: our method, LAP: Laplace)

where $\mathcal{A}$ is the set of final task assignment (*user, task*) pairs, and $d(u,t)$ is the Euclidean distance (in km) between selected worker u and the task t. Note that the distance can be changed to other metrics, such as Manhattan distance and map route distance, according to the practical use cases.

5.2 Results on Simulation

The evaluation is conducted with six tunable parameters (see Table 1) on the simulated $n \times n$ grid-cell target area. By alternatingly tuning one of these parameters while fixing the others, we study how our framework performs under different settings. For each parameter setting, we repeat 1000 trials and record the mean ATD. The evaluation results reported in Figure 3 show that our mechanism generally can reduce ATD by up to 45% compared to Laplace.

In particular, we observe that a smaller ATD can be achieved for MCS task allocation either by increasing the number of candidates (Figure 3a), downsizing the target area (Figure 3b), or loosening the privacy level (Figure 3c). Compared to Laplace, our method achieves significantly smaller ATD in all settings. More specifically, the utility loss (measured by ATD) incurred by our privacy-preserving method is only about half of Laplace. In addition, the increase of ATD difference between our method and Laplace when loosening the differential privacy level, shown in Figure 3c, indicates that a larger ϵ gives more search space for our framework to to approach the optimal solution.

Task number (Figure 3d), task spatial distribution (Figure 3e) and candidate spatial distribution (Figure 3f) are also considered as parameters for evaluation. Besides uniform, we also inspect distributions around the center and corner (Figure 3g)[6] are the inspected distributions. The generally consistent ATD values of our method shown in Figure 3d, 3e and 3f elucidate that our method has stable performance regarding the task number, the task distributions, or the candidate spatial distributions. More importantly, our mechanism always obtains much smaller ATD than Laplace across different settings.

[6]A dark grid has 9× probability larger than a white grid to be a task or candidate location.

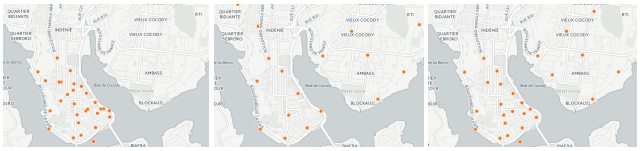


Figure 4: Task distributions in D4D dataset.

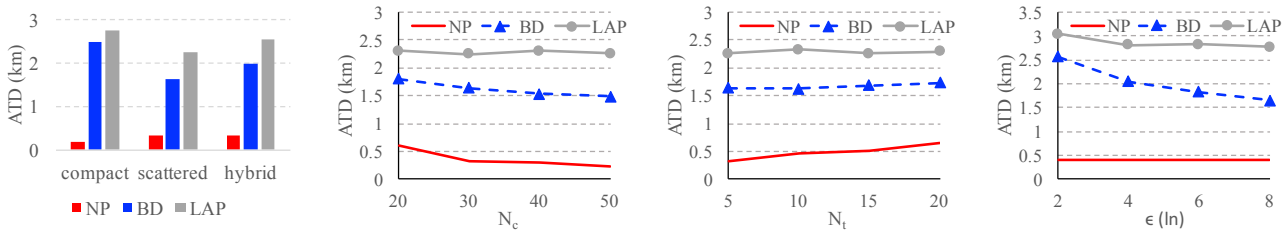
Note that in Figure 3f where the candidate distribution is not uniform, we also show ATD of our method when still supposing uniform candidate distribution during the optimization. We can observe that the inconsistent candidate distribution assumption will lower the performance of our method to some extent. Therefore, an accurate candidate distribution estimation is necessary in real-life deployment.

5.3 Results on D4D

Similar to [15], we use the cell tower positions in Abidjan as the total set of locations $\mathcal{L}$ and consider three types of task distributions, *compact*, *scattered*, and *hybrid*, which are shown in Figure 4 (default: scattered). We use 10:00-19:00 in workdays as the experimental period. Every one hour, the MCS platform needs to do one round of task allocation. In each round of task allocation, the task number ranges from 5 to 20 (default: 5), and the candidate number ranges from 20 to 50 (default: 30). Note that for each one-hour time slot, we learn a separate candidate distribution π according to candidates' uploaded obfuscated locations. The total task period lasts for two weeks, i.e., 10 workdays. The privacy level ϵ ranges from ln(2) to ln(8) (default: ln(4)).

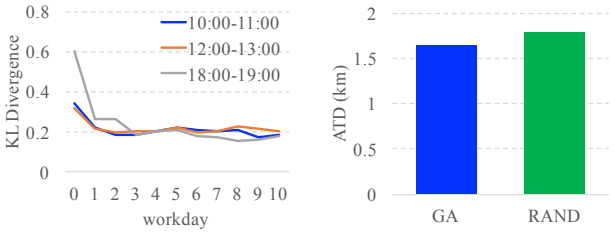
Figure 5 shows the evaluation results. Generally, the results are similar to the simulation results and our method can always achieve a smaller ATD than Laplace. In the following, we first investigate the impact of two important factors, i.e., geo-distribution estimation and GA-based initialization, on the performance of our method. We then evaluate its runtime performance.

Geo-distribution Estimation. To evaluate the effectiveness of our geo-distribution estimation (Sec. 4.3), we measure the difference of our estimated π' and the actual π^* using



(a) varying task distribution (b) varying candidate number (c) varying task number (d) varying privacy level

Figure 5: Evaluation results on D4D. (NP: no-privacy, BD: our method, LAP: Laplace)



(a) Geo-distribution Estimation (b) Genetic Algorithm

Figure 6: Submodule evaluation on D4D.

Kullback-Leibler divergence [18], which can quantify how much information is lost if using π' to represent π^*:

$$D_{KL}(\pi' || \pi^*) = \sum_{l \in \mathcal{L}} \pi'(l) \log \frac{\pi'(l)}{\pi^*(l)} \tag{31}$$

The more similar π' and π^* are, the lower D_{KL} is. Figure 6a shows the D_{KL} for three example one-hour time slots, and we set the initial value of π' to the uniform distribution. We can see that after two or three days, D_{KL} can be reduced to about 0.2, which is much smaller than the initial D_{KL} (i.e. π' is uniform), indicating the effectiveness of our geo-distribution estimation method.

Genetic Algorithm-based Initialization. To verify the effectiveness of GA-based initialization (Sec. 4.2), we compare it with random selection of the initial value of $\hat{x}_0$. As shown in Figure 6b, GA-based initialization can effectively reduce 10% of ATD compared to random initialization.

Runtime Performance. We use MOSEK 7.1 [1] to solve our linear optimization problems. On our test PC (Intel core i7-3612QM, 8GB RAM), it takes about 23.6 and 0.2 seconds to do one round of geo-obfuscation function generation and obfuscation-aware task allocation, respectively. Hence, compared to no-privacy task allocation, our framework introduces an overhead of less than 30 seconds, which is totally acceptable in real-life MCS applications.

6. RELATED WORK

We review the related work from the following two aspects in MCS literature: *task allocation* and *location privacy*.

6.1 Task Allocation in MCS

The objective of task allocation in MCS is to optimize the overall system utility while completing all (or a high percentage of) the tasks in the target sensing area. In the current literature, such system utilities can be roughly classified into four categories: 1) *sensing data quality* [29, 28, 36], which tries to maximize the data quality measured by a certain metric (mostly used in environmental monitoring tasks); 2) *incentive cost* [35, 32], which aims at minimizing the total budget (from the task organizer perspective) for an MCS task with different incentive mechanisms, such as pay per participant [35] or pay per task [32]; 3) *energy consumption* [25, 33], whose objective is to identify an optimal collaborative data sensing and uploading scheme with energy-saving techniques such as piggybacking [33]; 4) *travel distance* [15, 16, 20], where the travel distance of a user for accomplishing a task is considered in task allocation, in order to minimize the overall travel distance for all the tasks.

In this study, we advocate for the utility of minimizing travel distance, as it is a critical issue for both participants (i.e., users may not be willing to accomplish a task at the expense of a long-distance travel) and task organizers (i.e., an organizer will not appreciate a delayed sensing result caused by a participant's long traveling time). The other kinds of utility metrics, such as the monetary budget under certain incentive mechanisms, will be studied in our future work.

6.2 Location Privacy in MCS

Location privacy in MCS has attracted increasing research interests. Based on a recent survey on the MCS privacy issues [22], cloaking is still a widely used strategy in practice for protecting location privacy, e.g., [10, 23, 27]. However, these works all have the same drawback of being sensitive to the adversary's prior knowledge. In order to avoid this issue, differential privacy starts to be introduced in MCS. Wang et al. [30] proposed to leverage differential geo-obfuscation in environment monitoring tasks, whose utility is measured by the overall sensing error of the target area. Our work, by using the metric of travel distance, is not limited to environment monitoring tasks. A closely related work to ours is presented in [26], which also attempted to optimize workers' travel distance under differential privacy protection. However, their mechanism needs a third-party trustful entity to first collect users' real locations before perturbation. They proposed to let users' cellular service providers act as such a third-party, but how to incentivize service providers for participation is a hard issue in practice. In our solution, we let mobile users obfuscate their locations directly on their smartphones, thus avoiding such a trustful third-party.

7. CONCLUSION

This paper addresses the privacy-preserving problem in MCS task allocation. It uses differential geo-obfuscation to protect users' location privacy regardless of adversaries' prior knowledge, without the involvement of any trustful third-party. Meanwhile, it aims at minimizing users' travel distance. To this end, this paper proposes a mixed-integer

nonlinear program to collectively optimize both differential geo-obfuscation and task allocation using Benders Decomposition. The proposed privacy-preserving solution is verified on both simulation and real-world user mobility traces.

In the future, we plan to consider limiting each individual worker's travel distance when minimizing the average travel distance for all selected workers, as a long travel distance may discourage a worker's motivation for completing a task. Moreover, we will evaluate our framework on a finer-grained user mobility dataset, such as user activity data in location based social networks with exact GPS coordinates.

8. ACKNOWLEDGMENTS

This project is partially supported by NSFC Grant no. 61572048 and 71601106, State Language Commission Key Program Grant no. ZDI135-18, Hong Kong ITF Grant no. ITS/391/15FX, and ERC Consolidator Grant no. 683253 (GraphInt).

9. REFERENCES

[1] MOSEK. `https://www.mosek.com/`, 2016. Accessed: 2016-10-17.

[2] WAZE - Google Play. `https://play.google.com/store/apps/details?id=com.waze&hl=en`, 2016. Accessed: 2016-10-17.

[3] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *CCS*, pages 901–914, 2013.

[4] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and A. Mahajan. Mixed-integer nonlinear optimization. *Acta Numerica*, 22:1–131, 2013.

[5] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik*, 4(1):238–252, 1962.

[6] V. D. Blondel, M. Esch, C. Chan, F. Clérot, P. Deville, E. Huens, F. Morlot, Z. Smoreda, and C. Ziemlicki. Data for development: the d4d challenge on mobile phone data. *arXiv preprint arXiv:1210.0137*, 2012.

[7] N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. Optimal geo-indistinguishable mechanisms for location privacy. In *CCS*, pages 251–262, 2014.

[8] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[9] Y. Chon, N. D. Lane, F. Li, H. Cha, and F. Zhao. Automatically characterizing places with opportunistic crowdsensing using smartphones. In *UbiComp*, pages 481–490, 2012.

[10] C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos. Anonysense: privacy-aware people-centric sensing. In *MobiSys*, pages 211–224, 2008.

[11] A. K. Dey. Understanding and using context. *PUC*, 5(1):4–7, 2001.

[12] C. Dwork. Differential privacy: A survey of results. In *TAMC*, pages 1–19, 2008.

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

[14] A. M. Geoffrion. Generalized benders decomposition. *Journal of optimization theory and applications*, 10(4):237–260, 1972.

[15] B. Guo, Y. Liu, W. Wu, Z. Yu, and Q. Han. Activecrowd: A framework for optimized multitask allocation in mobile crowdsensing systems. *IEEE THMS*, 2016.

[16] S. He, D.-H. Shin, J. Zhang, and J. Chen. Toward optimal allocation of location dependent tasks in crowdsensing. In *INFOCOM*, pages 745–753, 2014.

[17] K.-R. Koch. *Introduction to Bayesian statistics*. Springer Science & Business Media, 2007.

[18] S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[19] A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems. *Econometrica: Journal of the Econometric Society*, pages 497–520, 1960.

[20] Y. Liu, B. Guo, Y. Wang, W. Wu, Z. Yu, and D. Zhang. Taskme: multi-task allocation in mobile crowd sensing. In *UbiComp*, 2016.

[21] M. Mitchell. *An introduction to genetic algorithms*. MIT press, 1998.

[22] L. Pournajaf, D. A. Garcia-Ulloa, L. Xiong, and V. Sunderam. Participant privacy in mobile crowd sensing task management: A survey of methods and challenges. *ACM SIGMOD Record*, 44(4):23–34, 2016.

[23] L. Pournajaf, L. Xiong, V. Sunderam, and S. Goryczka. Spatial task assignment for crowd sensing with cloaked locations. In *MDM*, volume 1, pages 73–82, 2014.

[24] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu. Ear-phone: an end-to-end participatory urban noise mapping system. In *IPSN*, pages 105–116, 2010.

[25] X. Sheng, J. Tang, and W. Zhang. Energy-efficient collaborative sensing with mobile phones. In *INFOCOM*, pages 1916–1924, 2012.

[26] H. To, G. Ghinita, and C. Shahabi. A framework for protecting worker location privacy in spatial crowdsourcing. *Proc. VLDB Endowment*, 7(10):919–930, 2014.

[27] I. J. Vergara-Laurens, D. Mendez, and M. A. Labrador. Privacy, quality of information, and energy consumption in participatory sensing systems. In *PerCom*, pages 199–207, 2014.

[28] L. Wang, D. Zhang, A. Pathak, C. Chen, H. Xiong, D. Yang, and Y. Wang. Ccs-ta: quality-guaranteed online task allocation in compressive crowdsensing. In *UbiComp*, pages 683–694, 2015.

[29] L. Wang, D. Zhang, Y. Wang, C. Chen, X. Han, and A. M'hamed. Sparse mobile crowdsensing: challenges and opportunities. *IEEE Communications Magazine*, 54(7):161–167, 2016.

[30] L. Wang, D. Zhang, D. Yang, B. Y. Lim, and X. Ma. Differential location privacy for sparse mobile crowdsensing. In *ICDM*, 2016.

[31] L. Willenborg and T. De Waal. *Elements of statistical disclosure control*, volume 155. Springer Science & Business Media, 2012.

[32] H. Xiong, D. Zhang, G. Chen, L. Wang, and V. Gauthier. Crowdtasker: Maximizing coverage quality in piggyback crowdsensing under budget constraint. In *PerCom*, pages 55–62, 2015.

[33] H. Xiong, D. Zhang, L. Wang, and H. Chaouchi. Emc 3: Energy-efficient data transfer in mobile crowdsensing under full coverage constraint. *IEEE TMC*, 14(7):1355–1368, 2015.

[34] D. Zhang, L. Wang, H. Xiong, and B. Guo. 4w1h in mobile crowd sensing. *IEEE Communications Magazine*, 52(8):42–48, 2014.

[35] D. Zhang, H. Xiong, L. Wang, and G. Chen. Crowdrecruiter: selecting participants for piggyback crowdsensing under probabilistic coverage constraint. In *UbiComp*, pages 703–714, 2014.

[36] Y. Zhu, Z. Li, H. Zhu, M. Li, and Q. Zhang. A compressive sensing approach to urban traffic estimation with probe vehicles. *IEEE TMC*, 12(11):2289–2302, 2013.