Title	Mobile Crowdsourcing Task Allocation with Differential-and-Distortion Geo-Obfuscation
Authors	Wang, Leye; Yang, Dingqi; Han, Xiao; Zhang, Daqing; Ma, Xiaojuan
Source	IEEE Transactions on Dependable and Secure Computing, v. 18, (2), March-April 2021, article number 8695792, p. 967-981
Version	Accepted Version
DOI	10.1109/tdsc.2019.2912886
Publisher	IEEE
Copyright	© 2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Mobile Crowdsourcing Task Allocation with Differential-and-Distortion Geo-Obfuscation

Leye Wang, Dingqi Yang, Xiao Han, Daqing Zhang, *Fellow, IEEE*, and Xiaojuan Ma

Abstract—In mobile crowdsourcing, organizers usually need participants' precise locations for optimal task allocation, e.g., minimizing selected workers' travel distance to task locations. However, the exposure of users' locations raises privacy concerns. In this paper, we propose a location privacy-preserving task allocation framework with geo-obfuscation to protect users' locations during task assignments. More specifically, we make participants obfuscate their reported locations under the guarantee of two rigorous privacy-preserving schemes, *differential* and *distortion* privacy, without the need to involve any third-party trusted entity. In order to achieve optimal task allocation with the differential-and-distortion geo-obfuscation, we formulate a mixed-integer non-linear programming problem to minimize the expected travel distance of the selected workers under the constraints of differential and distortion privacy. Moreover, a worker may be willing to accept multiple tasks, and a task organizer may be concerned with multiple utility objectives such as task acceptance ratio in addition to travel distance. Against this background, we also extend our solution to the multi-task allocation and multi-objective optimization cases. Evaluation results on both simulation and real-world user mobility traces verify the effectiveness of our framework. Particularly, our framework outperforms Laplace obfuscation, a state-of-the-art geo-obfuscation mechanism, by achieving up to 47 percent shorter average travel distance on real-world data under the same level of privacy protection.

Index Terms—Mobile crowdsensing, differential location privacy, distortion location privacy, task allocation, travel distance

◆

1 INTRODUCTION

Mobile CrowdSourcing (MCS) [1], also known as mobile crowdsensing [2], [3] and spatial crowdsourcing [4], has attracted lots of interest from academia [5]. Nowadays, industry has also seen many successful MCS applications such as TaskRabbit.[1]

On a typical MCS platform, users are registered as candidate workers. When new MCS tasks come, the platform selects a proper subset of candidates to complete the tasks by providing them with some monetary 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. If the travel distance is too long, participants will probably be unwilling to conduct the task (i.e., the task acceptance probability will be lowered). Therefore, following previous work [6], [7], we consider minimizing the overall travel distance of workers in 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 indicates that users' *location privacy* is at risk. Note that in task allocation, while only a subset of candidates are selected as workers, 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 [8], most existing mechanisms (e.g., [9], [10], [11]) employ a cloaking-based idea (i.e., using a coarse area to represent a precise location) to provide location protection. However, their expected privacy guarantee can be easily downgraded if adversaries hold certain prior knowledge [12]. For example, if an adversary foreknows that a user is a student, and the cloaking area includes both a school and government

---

[1] https://www.taskrabbit.com

- *L. Wang is with the Key Lab of High Confidence Software Technologies, Ministry of Education, and Software Institute, Peking University, Beijing, Shi 100871, China. E-mail: leyewang@pku.edu.cn.*
- *D. Zhang is with the Key Lab of High Confidence Software Technologies, Ministry of Education, and Software Institute, Peking University, Beijing, Shi 100871, China, and also with Telecom SudParis, France. E-mail: dqzhang@sei.pku.edu.cn.*
- *D. Yang is with the eXascale Infolab, University of Fribourg, Fribourg 1700, Switzerland. E-mail: dingqi@exascale.info.*
- *X. Han is with the Shanghai University of Finance and Economics, Yangpu Qu, Shanghai Shi 200433, China. E-mail: xiaohan@mail.shufe.edu.cn.*
- *X. Ma is with the Hong Kong University of Science and Technology, Hong Kong SAR, China. E-mail: mxj@cse.ust.hk.*

Manuscript received 21 May 2018; revised 11 Mar. 2019; accepted 8 Apr. 2019. Date of publication 23 Apr. 2019; date of current version 12 Mar. 2021.
(Corresponding author: Xiao Han.)
Digital Object Identifier no. 10.1109/TDSC.2019.2912886

office, the adversary can infer rather confidently 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. [7] 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 the motivation to participate.

(3) *Relying on a single privacy scheme*. Although a variety of location privacy-preserving schemes have been proposed in current literature, existing location protection mechanisms for MCS task allocation usually adopt a single scheme [8]. Since each protection scheme has its own assumptions and objective, a mechanism incorporating multiple privacy schemes would be more desired in reality, as it can provide more robust protection.

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* [13] to provide theoretically guaranteed protection regardless of adversaries' prior knowledge. Consequently, some Location-Based Services (LBS) have proposed several *differential geo-obfuscation* mechanisms [12], [14]. 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 arbitrary 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 trusted third-parties.

However, differential privacy only bounds the *relative information gain*, i.e., *the difference between adversaries' posterior knowledge after observing the obfuscated location and their prior knowledge*. We still do not have a clear idea about adversaries' *absolute information gain* of their target users, e.g., the distance between adversaries' guessed locations and users' actual positions. To this end, we consider leveraging *distortion privacy* [15] along with differential privacy in geographic obfuscation. Distortion privacy guarantees that the expected inference error of adversaries is larger than a threshold under an arbitrary inference attack on locations, assuming that adversaries only hold an overall prior knowledge of users.[2]

Optimizing MCS task allocation under differential-and-distortion geo-obfuscation needs to address the following challenges. In LBS, each individual user's geo-obfuscation method can be optimized independently by considering only his/her own actual and obfuscated locations [14]. However, 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 challenging to optimally incorporate differential-and-distortion geo-obfuscation in MCS task allocation while minimizing the workers' overall travel distance. Besides, a user may be willing to accept multiple tasks [6], and an MCS organizer may also be concerned with multiple utility metrics, e.g., task acceptance ratio as well as travel distance [7]. Hence, a practical privacy-preserving solution should also be able to cover these scenarios.

In this paper, we propose an MCS task allocation framework to protect participants' location privacy with differential-and-distortion geo-obfuscation, while minimizing selected workers' overall travel distance. This framework is further extended to multi-task allocation and multi-objective optimization scenarios. The contributions of this paper are summarized as follows.

(1) To the best of our knowledge, this is the first work to introduce differential and distortion geo-obfuscation jointly to MCS task allocation.

(2) To minimize candidate workers' travel distance, we propose an optimal privacy-preserving MCS task allocation framework with two interleaved modules: *differential-and-distortion geo-obfuscation* and *obfuscation-aware task allocation*. We then formulate a *mixed-integer nonlinear program (MINLP)* to optimize the two aforementioned modules collectively for travel distance minimization (Section 3). As directly solving MINLP is hard, we propose a method integrating *Benders Decomposition* [17], *Genetic Algorithm* and *Bayesian Analysis* techniques to obtain the solution (Section 4). Furthermore, we extend our approach to *multi-task allocation* and *multi-objective optimization* (Section 5).

(3) The evaluation on both simulated and real-world user mobility traces verifies that our framework can reduce up to 47 percent of travel distance of selected workers compared to the state-of-the-art geo-obfuscation mechanism, Laplace obfuscation [12], under the same level of differential and distortion privacy protection.

2 BACKGROUND

2.1 Mobile Crowdsourcing Task Model

In MCS, there are two task assignment models [7], *Worker Selected Task* (WST) and *Server Assigned Task* (SAT). In the WST model, the MCS platform publishes tasks online and candidates can select any tasks to conduct without exposing their location information. In the SAT model, candidates upload their locations to the platform and the platform selects some candidates to allocate tasks. Although the WST model is more friendly to users' privacy, it falls shorts in not being able to globally control the task allocation process. In contrast, the

---

[2] To quantify adversaries' absolute information gain, certain assumptions on the prior knowledge are required [15], [16].

SAT model can better optimize the overall task efficiency as the platform has the whole knowledge of all the candidates' locations. Our research attempts to combine the advantages of both models, i.e., using the SAT model to get the good running performance of all the MCS tasks, while still protecting users' location privacy.

In this paper, we assume that the number of tasks is smaller than that of candidate workers during the assignment, as this is more common in practical MCS platforms. For instance, the worker number of TaskRabbit was more than ten times of the task number (per day) in 2011.[3] It is also worth noting that our solution can be modified to the case when the task number is larger than the worker number, which will be discussed in Section 8.

2.2 Differential Geo-Obfuscation

Differential privacy is recently introduced in location protection by Andres et al. [12]. It performs as a probabilistic geo-obfuscation process, i.e., a user first obfuscates his/her 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, supposing 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^*.

X*Differential Privacy* [12], [14]. *Suppose the concerned area includes a set of locations $\mathcal{L}$, then a probabilistic geo-obfuscation function P satisfies ϵ-differential-privacy, iff*

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

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 to reflect the intuition that if l_1 and l_2 are near (i.e., small $d(l_1, l_2)$), they should be more indistinguishable. Note that $\mathcal{L}$ 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) [14]. While $d(l_1, l_2)$ could be any distance metric, usually it is considered to be euclidean distance with the unit of kilometer [14].

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 knowledge 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 [12]. Thus, differential geo-obfuscation can robustly protect users' location privacy against adversaries with arbitrary prior knowledge. Please refer to [12] for the theoretical proof.

2.3 Distortion Geo-Obfuscation

Distortion location privacy is another rigorous location protection scheme for limiting the expected inference error of adversaries larger than a pre-defined threshold. The protection is ensured with the assumption that adversaries only foreknow users' overall location distribution.[4] This is achieved by first modeling an *optimal* attack σ^* that minimizes the expected inference error [16]

$$\arg\min_{\sigma^*} \sum_{l \in \mathcal{L}} \pi(l) \sum_{l^* \in \mathcal{L}} P(l^*|l) \sum_{\hat{l} \in \mathcal{L}} \sigma^*(\hat{l}|l^*) d(\hat{l}, l), \tag{2}$$

where π is users' overall location distribution. Then, we bound the expected inference error of the optimal attack σ^* larger than a threshold δ. The formal definition is:

Distortion Privacy [15]. *A probabilistic geo-obfuscation function P satisfies δ-distortion-privacy iff*

$$\sum_{l \in \mathcal{L}} \pi(l) \sum_{l^* \in \mathcal{L}} P(l^*|l) \sum_{\hat{l} \in \mathcal{L}} \sigma^*(\hat{l}|l^*) d(\hat{l}, l) \geq \delta, \tag{3}$$

where δ indicates the user's privacy requirement of the lower bound of the expected optimal attack inference error.

While the above definition gives a rigorous formulation of distortion privacy, it is hard to directly apply, as the optimal attack σ^* in the definition requires solving the optimization problem (2). Fortunately, existing literature has proved that P satisfying the following two constraints are equivalent to distortion privacy [15]

$$\sum_{l \in \mathcal{L}} \pi(l) P(l^*|l) d(\hat{l}, l) \geq y(l^*), \ \forall \hat{l}, l^* \in \mathcal{L} \tag{4}$$

$$\sum_{l^* \in \mathcal{L}} y(l^*) \geq \delta. \tag{5}$$

Both as geo-obfuscation methods, differential and distortion privacy can work together to provide more comprehensive privacy protection. Differential privacy limits the relative information gain of adversaries regardless of their prior knowledge, whereas distortion privacy bounds the absolute information gain with a moderate assumption that adversaries only foreknow the overall location distribution of participants.

It is worth noting that the protection effect of differential-and-distortion privacy depends only on whether the geo-obfuscation function P satisfies the definitions (1) and (3). Hence, it is flexible to protect many types of attacks in reality. For example, MCS workers may face the attacks incurred by fake tasks sent by task owners or platforms.[5] But as long as a user adopts the differential-and-distortion geo-obfuscation function to obfuscate his/her location, the privacy protection effect can stand no matter whether the task is fake or not.

---

[3] *'... Today the site, since renamed TaskRabbit, has more than 1,500 runners (a.k.a., workers) in San Francisco, Boston, Los Angeles, and Orange County fulfilling up to 3,000 tasks per month ...'* — https://www.wired.com/2011/07/mf_taskrabbit/

[4] We need to make this assumption for distortion privacy. For example, instead of the overall distribution of participants, if an adversary happens to know a victim's exact location from some auxiliary data sources (i.e., $\pi^*(r) = 1$ where r is the victim's true location), then the inference error will always be *zero*, i.e., no distortion privacy can be obtained [15].

[5] https://splinternews.com/if-you-use-waze-hackers-can-stalk-you-1793856445

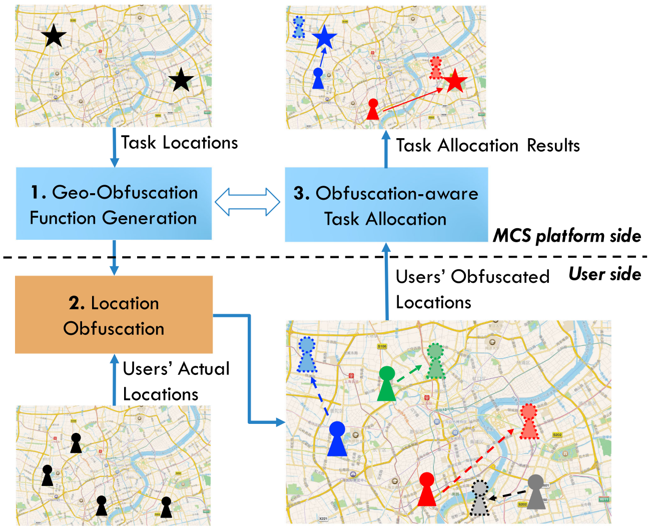


Fig. 1. Workflow of task allocation with geo-obfuscation.

3 PROBLEM ANALYSIS

In this section, we first illustrate the overall process of MCS task allocation with 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 tasks in a city which require workers to conduct. To protect users' privacy, rather than frequently requiring location updates, 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), 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 the appropriate ones. Selected workers can decide whether to accept the assigned tasks or not depending on their actual distance from the designated task locations.

In brief, the above process includes three steps, as shown in Fig. 1: (i) *Platform-side Geo-Obfuscation Function Generation*, (ii) *User-side Location Obfuscation*, and (iii) *Platform-side Obfuscation-aware Task Allocation*.

(i) *Platform-side Geo-Obfuscation Function Generation*. Before collecting candidates' locations, a probabilistic obfuscation function needs to be generated for candidates with certain differential and distortion 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 both differential and distortion privacy is guaranteed assuming that the adversary knows the obfuscation function [12], [15]. Hence, users can get privacy protection without needing to trust the platform, even the platform generates the obfuscation function.

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

(iii) *Platform-side Obfuscation-aware Task Allocation*. Finally, after receiving candidates' obfuscated locations, the MCS platform will assign tasks to proper workers, attempting to minimize selected workers' travel distance to the task locations. Since users' uploaded locations are obfuscated, directly taking them as actual locations and assigning tasks accordingly may not perform well. Instead, the obfuscation function should 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 two key problems in the above process: *differential-and-distortion geo-obfuscation* and *obfuscation-aware task allocation*.

3.2.1 Differential-and-Distortion Geo-Obfuscation

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

minimize: *Travel distance of selected workers*
subject to:

(i) *P satisfies differential privacy*
(ii) *P satisfies distortion privacy*

As differential and distortion privacy constraints have been given in Eqs. (1) and (4)-(5), respectively, we still need to model the objective: travel distance of selected users.

Objective: *Minimize Travel Distance*

To compute the expected travel distance of the selected workers, 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

$$d^*(l^*, l_t) = \frac{\sum_{l \in \mathcal{L}} \pi(l) P(l^*|l) d(l, l_t)}{\sum_{l \in \mathcal{L}} \pi(l) P(l^*|l)}, \tag{6}$$

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 Section 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 obfuscated l^*. Based on x, we can calculate the mean expected travel distance ($\mathbb{E}_{TD}$) of each selected user as

$$\mathbb{E}_{TD} = \sum_{l^* \in \mathcal{L}} \sum_{l_t \in \mathcal{L}} \frac{d^*(l^*, l_t) x(l^*, l_t)}{N_t} \tag{7}$$

$$= \sum_{l^* \in \mathcal{L}} \sum_{l_t \in \mathcal{L}} \frac{\sum_{l \in \mathcal{L}} \pi(l) P(l^*|l) d(l, l_t)}{\sum_{l \in \mathcal{L}} \pi(l) P(l^*|l) N_t} x(l^*, l_t). \tag{8}$$

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 $\mathbb{E}_{TD}$, this hypothetical x is also a variable to be optimized, i.e., P and x should be the best combination to achieve the optima. We use $\hat{x}$ to denote the x in the best combination $\{P, x\}$.

Then, given the number of tasks at each location l, denoted as $N_t(l)$, and the total number of candidates N_c,[6] we can mathematically formalize the problem of optimizing geo-obfuscation function P as

$$\min_{P, \hat{x}} \sum_{l^* \in \mathcal{L}} \sum_{l_t \in \mathcal{L}} \frac{\sum_{l \in \mathcal{L}} \pi(l) P(l^*|l) d(l, l_t)}{\sum_{l \in \mathcal{L}} \pi(l) P(l^*|l) N_t} \hat{x}(l^*, l_t), \tag{9}$$

s.t.

$$P(l^*|l_1) \leq e^{\epsilon d(l_1, l_2)} P(l^*|l_2) \qquad l_1, l_2, l^* \in \mathcal{L} \tag{10}$$

$$\sum_{l \in \mathcal{L}} \pi(l) P(l^*|l) d(\hat{l}, l) \geq y(l^*) \qquad \forall \hat{l}, l^* \in \mathcal{L} \tag{11}$$

$$\sum_{l^* \in \mathcal{L}} y(l^*) \geq \delta \tag{12}$$

$$\sum_{l^* \in \mathcal{L}} \hat{x}(l^*, l_t) = N_t(l_t) \qquad l_t \in \mathcal{L} \tag{13}$$

$$N_t = \sum_{l_t \in \mathcal{L}} N_t(l_t) \tag{14}$$

$$\sum_{l \in \mathcal{L}} \pi(l) P(l^*|l) = \pi(l^*) \qquad l^* \in \mathcal{L} \tag{15}$$

$$\sum_{l_t \in \mathcal{L}} \hat{x}(l^*, l_t) \leq \pi(l^*) N_c \qquad l^* \in \mathcal{L} \tag{16}$$

$$\sum_{l^* \in \mathcal{L}} P(l^*|l) = 1 \qquad l \in \mathcal{L} \tag{17}$$

$$P(l^*|l) \geq 0 \qquad l, l^* \in \mathcal{L} \tag{18}$$

$$\hat{x}(l^*, l_t) \in \mathcal{Z}_{\geq 0} \qquad l^*, l_t \in \mathcal{L}. \tag{19}$$

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. Constraint (10) is differential privacy; constraints (11) and (12) represent distortion privacy; constraint (13) guarantees that every task is assigned to a worker; constraint (15) ensures that the geo-obfuscation does not change candidates' overall location distribution;[7] constraint (16) ensures that the number of tasks assigned to users at l^* is smaller than the total number of users there; [8] constraint (17) and (18) are two general probability restrictions; constraint (19) ensures that the number of task allocations should be an integer.

As there is an integral constraint (19) and the objective function (9) is non-linear, this optimization is a *mixed-integer non-linear program* [19]. While state-of-the-art non-linear optimization techniques can deal with *convex* objectives effectively [20], unfortunately our objective function is non-convex. To this end, a specialized algorithm is needed to solve this MINLP for getting an effective geo-obfuscation function, which will be presented in Section 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 use $\tilde{x}$ to denote such a real task allocation scheme, and the problem of optimizing $\tilde{x}$ is formalized as

$$\min_{\tilde{x}} \sum_{l^* \in \mathcal{L}} \sum_{l_t \in \mathcal{L}} \frac{\sum_{l \in \mathcal{L}} \pi(l) P(l^*|l) d(l, l_t)}{\sum_{l \in \mathcal{L}} \pi(l) P(l^*|l) N_t} \tilde{x}(l^*, l_t), \tag{20}$$

s.t.

$$\sum_{l^* \in \mathcal{L}} \tilde{x}(l^*, l_t) = N_t(l_t) l_t \in \mathcal{L} \tag{21}$$

$$\sum_{l_t \in \mathcal{L}} \tilde{x}(l^*, l_t) \leq N_c(l^*) \qquad l^* \in \mathcal{L} \tag{22}$$

$$\tilde{x}(l^*, l_t) \in \mathcal{Z}_{\geq 0} \qquad l^*, l_t \in \mathcal{L}, \tag{23}$$

where $N_c(l^*)$ is the actual number of users with obfuscated location l^*. Now P is known and the only variable is $\tilde{x}$. Hence, this is a *mixed-integer linear program* (MILP). Solving MILP is easier than MINLP, and many off-the-shelf optimization tools can solve it with well-studied optimization techniques such as *branch and bound* [21]. Based on $\tilde{x}(l^*, l_t)$, which points out how many candidates at obfuscated l^* will be selected to conduct the tasks at l_t, we can then randomly select this number of workers from all the candidates reporting their locations as l^*.

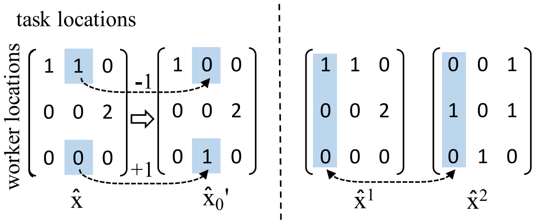
Note that the aforementioned optimized task allocation assumes that each user takes at most one task. In reality, users may be willing to accept multiple tasks and this provides extra opportunities to reduce workers' travel distance. In Section 5.1, we will extend our method to allow assigning multiple tasks to one candidate worker.

---

[6] We can get N_c by sending messages to all the users on the platform and collect their feedbacks before generating the geo-obfuscation function.

[7] Keeping key statistics invariant in obfuscation is a common practice in statistical disclosure control with many benefits [18]. In our case, for instance, this ensures that directly plotting candidates' obfuscated locations on the map can still roughly reflect user distribution. Such a map is usually an important part of the user interface for MCS applications (e.g., WAZE).

[8] Because we cannot foreknow the number of users whose obfuscated location is l^*, here we estimate it using the overall geographic distribution and the total number of candidates.



(a) Mutation (b) Crossover

Fig. 2. Illustrative examples of mutation and crossover.

4 GEO-OBFUSCATION OPTIMIZATION

To solve MINLP (9), we propose a method integrating the techniques such as Benders Decomposition (BD) [17], Genetic Algorithm (GA), and Bayesian analysis.

4.1 Benders Decomposition

The basic idea of BD is *divide-and-conquer* [22], 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 naturally includes two subsets of variables, P and $\hat{x}$, we can 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:

$$\min_{P} \sum_{l^* \in \mathcal{L}} \sum_{l_t \in \mathcal{L}} \frac{\sum_{l \in \mathcal{L}} \pi(l) P(l^*|l) d(l, l_t)}{\sum_{l \in \mathcal{L}} \pi(l) P(l^*|l) N_t} \hat{x}(l^*, l_t), \tag{24}$$

s.t.

$$P(l^*|l_1) \le e^{\epsilon d(l_1, l_2)} P(l^*|l_2) \qquad l_1, l_2, l^* \in \mathcal{L} \tag{25}$$

$$\sum_{l \in \mathcal{L}} \pi(l) P(l^*|l) d(\hat{l}, l) \ge y(l^*) \qquad \forall \hat{l}, l^* \in \mathcal{L} \tag{26}$$

$$\sum_{l^* \in \mathcal{L}} y(l^*) \ge \delta \tag{27}$$

$$\sum_{l \in \mathcal{L}} \pi(l) P(l^*|l) = \pi(l^*) \qquad l^* \in \mathcal{L} \tag{28}$$

$$\sum_{l^* \in \mathcal{L}} P(l^*|l) = 1 \qquad l \in \mathcal{L} \tag{29}$$

$$P(l^*|l) \ge 0 \qquad l, l^* \in \mathcal{L}. \tag{30}$$

By considering the Eq. (28), the objective (24) can be converted as follows:

$$\min_{P} \sum_{l^* \in \mathcal{L}} \sum_{l_t \in \mathcal{L}} \sum_{l \in \mathcal{L}} \frac{\pi(l) d(l, l_t)}{\pi(l^*) N_t} \hat{x}(l^*, l_t) P(l^*|l). \tag{31}$$

Given $\hat{x}$, the objective (31) is a linear function regarding P, and thus P-subproblem is a linear programming problem. Note that for implementation speedup, we adopt the δ-spanner-based approximation method to reduce the complexity of the ϵ-differential-privacy constraint (25) from $O(|\mathcal{L}|^3)$ to $O(|\mathcal{L}|^2)$; details can be found in [14].

x̂-*subproblem*:

$$\min_{\hat{x}} \sum_{l^* \in \mathcal{L}} \sum_{l_t \in \mathcal{L}} \frac{\sum_{l \in \mathcal{L}} \pi(l) P(l^*|l) d(l, l_t)}{\sum_{l \in \mathcal{L}} \pi(l) P(l^*|l) N_t} \hat{x}(l^*, l_t), \tag{32}$$

s.t.

$$\sum_{l^* \in \mathcal{L}} \hat{x}(l^*, l_t) = N_t(l_t) \qquad l_t \in \mathcal{L} \tag{33}$$

$$\sum_{l_t \in \mathcal{L}} \hat{x}(l^*, l_t) \le \pi(l^*) N_c \qquad l^* \in \mathcal{L} \tag{34}$$

$$\hat{x}(l^*, l_t) \in \mathcal{Z}_{\ge 0} \qquad l^*, l_t \in \mathcal{L}. \tag{35}$$

Given P, the objective (32) is a linear function regarding $\hat{x}$; considering the integral constraint (35), $\hat{x}$-subproblem is a mixed-integer linear programming problem.[9]

In a word, after BD, P-subproblem and $\hat{x}$-subproblem are both changed to (mixed-integer) linear programming problems, which can be efficiently solved with off-the-shelf tools. In our experiment, we find that usually after two or three 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 the Genetic Algorithm [23] to select the initial value of $\hat{x}_0$ that deserve testing based on the previously obtained local optima $\hat{x}$.[10] Based on the new $\hat{x}_0$, we can learn P, and followed by iterative BD for geo-obfuscation optimization. The key idea of GA is to generate a potential solution for utility testing from existing solutions with either *mutation* or *crossover* methods under a given probability, which is often set according to specific applications [23]. We design the mutation and crossover processes as follows (Fig. 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 $(l_3 \ne l_1)$. 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 $\hat{x}_0^{2'} = \hat{x}^2$.

---

[9] $\hat{x}$-subproblem is similar to the task allocation problem (Section 3.2.2) except for the difference between the constraints (22) and (34), as we do not know real user number in each obfuscated region when solving $\hat{x}$-subproblem.

[10] Using GA to construct a new feasible P is complicated due to the existence of the differential privacy constraint (25). We thus focus on generating new $\hat{x}_0$.

Note that for both mutation and crossover results, the constraint (34) 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 the constraint (34) stands after mutation or crossover. If it does not stand, we will re-run mutation or crossover until (i) the constraint (34) stands, or (ii) the re-running 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 π is hardly known, especially as candidates upload their obfuscated locations. Here, we propose to estimate π based on the obfuscated locations uploaded by candidates. In such a way, when a new round of task allocation starts, we can use an up-to-date approximation of π based on candidates' obfuscated locations in the 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 of 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 the incoming obfuscated locations that are reported by the candidates. This can be modeled using Bayesian analysis. Supposing a user's obfuscated location is l^* and the corresponding obfuscation function is P, we update π by the Bayes rule

$$\pi(l) \qquad \frac{\pi(l)P(l^*|l)}{\sum_{l' \in \mathcal{L}} \pi(l')P(l^*|l')}, \quad l \in \mathcal{L}. \tag{36}$$

In 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 [24]). With the continuously incoming observations (i.e., obfuscated locations), the estimated π will converge to the real candidates' geographic distribution, and the impact of π_0 on the estimated π is gradually reduced [25].

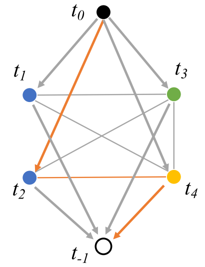
Note that this estimation has an implicit assumption that candidates' actual locations are sampled from the same hidden geographic distribution. In reality, users' mobility patterns could be affected by various *contexts* [26]; only under similar contexts, this assumption could stand. Hence, in implementation, we can estimate a set of π corresponding to various contexts (e.g., time, weekday/holiday [24]).

5 APPROACH EXTENSIONS

In this section, we further extend the proposed differential-and-distortion geo-obfuscation method from two aspects: *multi-task allocation* and *multi-objective optimization*.

5.1 Multi-Task Allocation

Assuming that a worker can do only one task, the mixed-integer linear program (20) makes an optimal task allocation given an obfuscation function P. In reality, a user may accept multiple tasks. Here, based on the task allocation scheme obtained by MILP (20), we investigate how to use multi-task allocation to further reduce workers' overall travel distance compared to single-task allocation.



The color of the task nodes indicates the location. Each path starting from node 't_0', ending at 't_{-1}' and not covering same color nodes is a possible task allocation. E.g., path $\langle t_0, t_2, t_4, t_{-1} \rangle$ indicates assigning a worker to both tasks t_2 and t_4 (more specifically, first finishing t_2 and then t_4).

Fig. 3. An illustrative example of task graph.

Motivated Example. Supposing tasks t_1 and t_2 are at location l_1 and l_2, respectively, Eq. (20) assigns t_1 to u_1 and t_2 to u_2. The expected travel distance of $u_1 \rightarrow l_1$ is 2 *km* and $u_2 \rightarrow l_2$ is 1 *km*. Then, the total distance is 3 *km*. If the distance between l_1 and l_2 is 1 *km* and a user can accept multiple tasks, then we can assign both t_1 and t_2 to u_2, leading to a shorter total distance of 2 *km* ($u_2 \xrightarrow{1\,\mathrm{km}} l_2 \xrightarrow{1\,\mathrm{km}} l_1$).

While multi-task allocation has been studied by previous MCS works [27], [28], [29], [30], they usually do not consider privacy protection. We then develop a method for multi-task allocation with differential-and-distortion geo-obfuscation. We elaborate more about why previous works cannot be directly applied to our scenario later in the related work section (Section 7.1).

Our proposed method converts the multi-task allocation problem into an MILP problem. To illustrate this conversion, we first build the task graph as follows. We create a start allocation node t_0 and an end allocation node t_{-1}; for each task $i \in \{1 \ldots N_t\}$, we create a node t_i (each t_i needs one worker; t_i and t_j could be at a same location). Then, we link t_0 to each t_i and each t_i to t_{-1}. The weight of the edge $\langle t_0, t_i \rangle$ is the expected travel distance of the assigned worker u_i to t_i according to MILP (20), while the weight of $\langle t_i, t_{-1} \rangle$ is set to zero. We also link the task nodes which may be co-allocated to one worker; the link weight is set to the distance between two task locations. Without the loss of generality, we suppose that multiple tasks in one location cannot be allocated to one user, but tasks in different locations can.[11] Hence, the tasks in one location will not be connected to each other. Fig. 3 shows an example of four tasks distributed in three locations. We use the node color to mark the location. Then, we define a valid *allocation path* as a path starting from t_0, ending at t_{-1} and not containing multiple same-color nodes. For example, $\langle t_0, t_2, t_4, t_{-1} \rangle$ is a valid allocation path; but $\langle t_0, t_2, t_4, t_1, t_{-1} \rangle$ is invalid. The sum of the edge weights is the expected travel distance for the worker to finish all the tasks in the path.

With the task graph, a task allocation strategy can be seen as a set of valid allocation paths covering every task node exactly once. In Fig. 3, the allocation result of MILP (20) is $\{\langle t_0, t_1, t_{-1} \rangle, \langle t_0, t_2, t_{-1} \rangle, \langle t_0, t_3, t_{-1} \rangle, \langle t_0, t_4, t_{-1} \rangle\}$ (four workers); while other task allocation results are also possible, e.g., $\{\langle t_0, t_1, t_{-1} \rangle, \langle t_0, t_2, t_4, t_{-1} \rangle, \langle t_0, t_3, t_{-1} \rangle\}$ (three workers). Then, our objective is to find a set of valid allocation paths with the

---

[11] It is common in practice that multiple tasks in one location cannot be finished by one user. For example, five tasks at one location need five different workers to sense the noise level to ensure data reliability. Also, if multiple tasks in one location can be finished by one user, these tasks can be combined into one 'big' task, and then our method still works.

smallest sum of weights, which can be formulated as the following MILP:

$$\min_{x} \sum_{i,j} w_{i,j} \cdot x_{i,j}, \tag{37}$$

s.t.

$$\sum_{i\in\{-1,0,1\ldots N_t\}} x_{i,j} = 1, \quad \forall j \in \{1 \ldots N_t\} \tag{38}$$

$$\sum_{j\in\{-1,0,1\ldots N_t\}} x_{i,j} = 1, \quad \forall i \in \{1 \ldots N_t\} \tag{39}$$

$$x_{i,j} + x_{j,i} \leq 1, \quad \forall i,j \in \{-1,0,1 \ldots N_t\} \tag{40}$$

$$\sum_{j\in\{1\ldots N_t\}} x_{0,j} \geq 1 \tag{41}$$

$$\sum_{j\in\{1\ldots N_t\}} x_{0,j} = \sum_{i\in\{1\ldots N_t\}} x_{i,-1} \tag{42}$$

$$x_{i,j} \in \{0,1\}, \quad \forall i,j \in \{-1,0,1 \ldots N_t\}, \tag{43}$$

where $x_{i,j}$ indicates whether the edge $\langle t_i, t_j \rangle$ exists in the allocation, and $w_{i,j}$ is the weight. Note that the solution of MILP (37) may still include two types of invalid paths, i.e., the path not starting (ending) at t_0 (t_{-1}), or the path covering multiple same-location tasks. We thus iteratively add more constraints and re-solve MILP (37) if an invalid path exists in the solution. The added constraint is

$$\sum_{\langle t_i,t_j\rangle\in IP} x_{i,j} \leq |IP| - 1, \tag{44}$$

where IP is an invalid path and $|IP|$ is its length. Theoretically, to eliminate all of the possible invalid paths, it needs to iteratively add at most $O(2^n)$ constraints. But in practice, the number is much smaller and the solution can be obtained for a moderate scale problem.

If the task number is very large, we can turn to the greedy algorithm for multi-task allocation, i.e., iteratively reallocating one task from a worker to another worker if this can reduce the total travel distance the most. Although the greedy algorithm may lead to a sub-optimal solution compared to MILP (37), its scalability is much better.

5.2 Multi-Objective Optimization

Besides travel distance, we may optimize multiple objectives simultaneously if we can also model other objectives mathematically. Here, we take another widely concerned utility metric, *acceptance ratio* [7], [31], as an example to show how to consider it together with travel distance.

Objective 2: *Maximize Acceptance Ratio*

We calculate the expected probability that a user whose obfuscation location is l^* would accept a task at l_t

$$p_{acc}(l^*, l_t) = \frac{\sum_{l\in\mathcal{L}} \pi(l)P(l^*|l)\mathcal{A}(l,l_t)}{\sum_{l\in\mathcal{L}} \pi(l)P(l^*|l)}, \tag{45}$$

where $\mathcal{A}(l,l_t)$ models the probability of a user at l accepting a task at l_t. As a variety of real-life MCS studies [32] have shown that users tend to accept tasks near them, we use a decreasing function of distance between l and l_t to model $\mathcal{A}$. In the literature, linear and power law distribution functions are two common methods to model $\mathcal{A}$ [7], [31].

Suppose N_t is the total number of task assignments and $x(l^*, l_t)$ denotes the number of task assignments which allocate the tasks at l_t to the users whose obfuscated location is l^*. Then, the expected acceptance ratio ($\mathbb{E}_{AR}$) of selected workers are

$$\mathbb{E}_{AR} = \sum_{l^*\in\mathcal{L}} \sum_{l_t\in\mathcal{L}} \frac{p_{acc}(l^*, l_t)x(l^*, l_t)}{N_t} \tag{46}$$

$$= \sum_{l^*\in\mathcal{L}} \sum_{l_t\in\mathcal{L}} \frac{\sum_{l\in\mathcal{L}} \pi(l)P(l^*|l)\mathcal{A}(l,l_t)}{\sum_{l\in\mathcal{L}} \pi(l)P(l^*|l)N_t} x(l^*, l_t). \tag{47}$$

Combining Two Objectives Together

Minimizing travel distance $\mathbb{E}_{TD}$ (8) and maximizing acceptance ratio $\mathbb{E}_{AR}$ (47) are two objectives that we aim to achieve simultaneously. To solve this bi-objective optimization problem, we can use *linear scaling* [33] to convert it to single-objective optimization, i.e.,

$$\min\ (1-\beta)\mathbb{E}_{TD} - \beta\mathbb{E}_{AR} \tag{48}$$

$$\Rightarrow \min \sum_{l^*\in\mathcal{L}} \sum_{l_t\in\mathcal{L}} \frac{\sum_{l\in\mathcal{L}} \pi(l)P(l^*|l)\mathcal{M}_\beta(l,l_t)}{\sum_{l\in\mathcal{L}} \pi(l)P(l^*|l)N_t} \hat{x}(l^*, l_t), \tag{49}$$

where $\mathcal{M}_\beta(l,l_t) = (1-\beta)d(l,l_t) - \beta\mathcal{A}(l,l_t)$.

Our previously proposed differential-and-distortion geo-obfuscation framework for minimizing travel distance can be directly applied to the objective (49). Specifically, by varying β from 0 to 1, we can make the trade-off between travel distance minimization and acceptance ratio maximization for getting a set of Pareto optimal solutions [33].

Setting of β. Generally, we need to set $\beta \in [0,1]$ in $\mathcal{M}_\beta$ to solve the optimization problem. But under certain conditions, we prove that β does not impact the solution.

Theorem 1. *If users' task acceptance model* $\mathcal{A}$ *is linear w.r.t. the distance away from the task location, and the minimum probability in* $\mathcal{A}$ *between any two locations is larger than zero, i.e.,* $\min\mathcal{A}(l,l') > 0$, $\forall l,l' \in \mathcal{L}$, *then minimizing* $\mathbb{E}_{TD}$ *is equivalent to maximizing* $\mathbb{E}_{AR}$.

The proof is in shown in Appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TDSC.2019.2912886. Theorem 1 suggests that if the acceptance probability follows a linear model and the target area is relatively small (i.e., the probability of a user accepting any task in the area is larger than zero), then minimizing travel distance and maximizing acceptance ratio is equivalent. In other words, the specific value of β does not impact the optimization in such cases.

6 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. Second, to validate its applicability in real-world use cases, we also verify our

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
δ	(LAP)	distortion privacy level (default is set to the same value as Laplace)
π	uniform	candidate worker spatial distribution
τ	uniform	task spatial distribution

framework on a real-life mobility dataset, D4D [34], which includes 50,000 users' two-week mobility traces represented by their mobile phone call logs.

6.1 Experiment Setup

6.1.1 Evaluation Scenarios

We run experiments with both simulations and real datasets.

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 1 km × 1 km. We vary several key parameters in Table 1 to evaluate our framework in different settings.

D4D dataset [34] includes 50,000 users' phone call traces in Ivory Costa, which is widely used to evaluate task allocation mechanisms in MCS [6], [27], [35]. Referring to [6], [27], 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 Ivory Costa, *Abidjan*, as the target area, and randomly distribute tasks to a group of cell towers within the area.

6.1.2 Evaluation Metric

Average Travel Distance (ATD). Referring to [36], we use the euclidean distance to measure the distance between workers and tasks

$$ATD = \sum_{(u,t)\in\mathcal{S}} d(u,t)/|\mathcal{S}|, \tag{50}$$

where $\mathcal{S}$ is the set of final task assignment (*user, task*) pairs, and $d(u,t)$ is the euclidean distance (in km) between the 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.

6.1.3 Baselines

Laplace. We compare our framework with the state-of-the-art differential geo-obfuscation mechanism [12] 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

$$P(l^*|l) \propto e^{-\epsilon\frac{d(l,l^*)}{D(\mathcal{L})}}, \tag{51}$$

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

Since Laplace only constrains differential privacy, to make it comparable to our method, given the differential privacy level ϵ, we calculate the 'actual' distortion privacy δ that Laplace achieves with Eqs. (4) and (5). Then, we use this δ in our optimization formulation.

No-Privacy. We show the optimal task allocation results when candidates' real locations are reported, to see the assignment performance loss incurred by privacy protection.

6.2 Results on Simulation

The evaluation is conducted with a set of tunable parameters (see Table 1) on the simulated $n \times n$ grid-cell target area. By alternatively tuning one of these parameters while fixing the others, we study how our framework performs under different settings. For each parameter setting, we repeat 100 trials and record the mean ATD. The evaluation results are reported in Fig. 4.

In particular, we observe that a smaller ATD can be often achieved for MCS task allocation either by increasing the number of candidates (Fig. 4a), reducing the number of tasks (Fig. 4d), downsizing the target area (Fig. 4c) or loosening the privacy level (Fig. 4b). Compared to Laplace, our method achieves significantly smaller ATD.

We also change task spatial distribution (Fig. 4e) and candidate worker spatial distribution (Fig. 4f) to see simulation results. Besides uniform, we inspect distributions around the center and corner (Fig. 5). Our method still consistently outperforms Laplace in obtaining lower ATD. Note that in Fig. 4f 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 inaccurate assumption about the candidates' distribution will lower the performance of our method, which can lead up to a 20 percent increase in ATD. Therefore, an accurate candidate distribution estimation is necessary for the real deployment, verifying the necessity of our proposed candidate geo-distribution estimation method in Section 4.3.

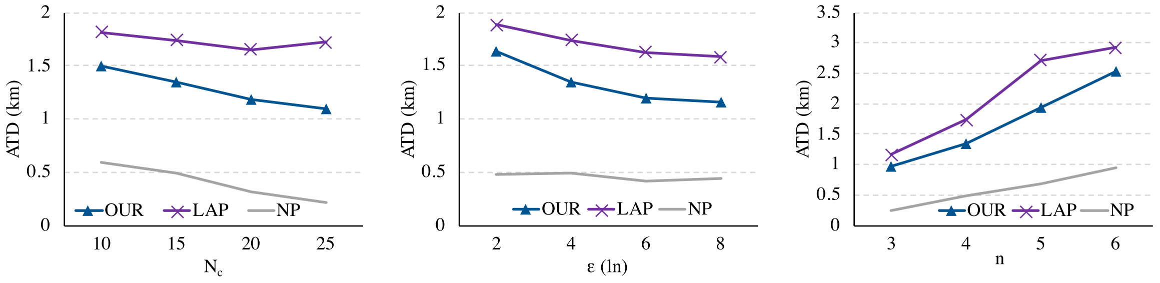
As *Benders Decomposition* and *Genetic Algorithm* are heuristic optimization methods which may not obtain the global optimal P^*, we compare the ATD of our obtained P to P^* in Fig. 6. Since calculating P^* is expensive,[12] we show the comparison when $N_t = 2$. Results show that our solution is very competitive to the optimal solution with similar ATD ($< 2\%$ of loss). More specifically, we compare our method to the approach which only leverages BD to obtain P (not using GA). We find that if only using BD, the solution quality is much worse, which leads to $> 30\%$ of loss compared to the optimal solution. This verifies that, by introducing GA, we significantly improve the solution quality in practice.

6.3 Results on Real Human Mobility Datasets

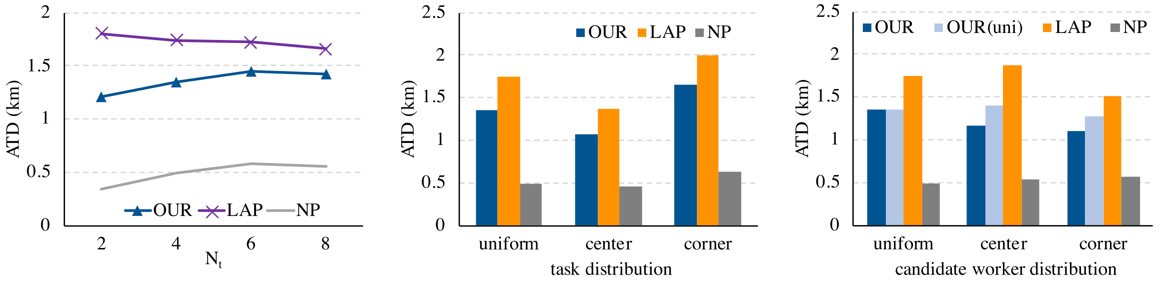
Now, we use a real-life human mobility dataset, D4D [34], to evaluate our method. Similar to [6], we use the cell tower positions in Abidjan as the total set of locations $\mathcal{L}$ and consider three types of task distributions, *compact*, *scattered*, and

---

[12] We exhaustively enumerate possible task allocation $\hat{x}$ and solve LP (24) to get a candidate set of $\{P\}$; P^* with the smallest objective is selected from $\{P\}$. This calculation needs $O(|\mathcal{L}|^{N_t})$ iterations.



(a) vary N_c (b) vary ϵ (c) vary n



(d) vary N_t (e) vary task distribution (f) vary candidate worker distribution

Fig. 4. Simulation results of average travel distance (NP: No-privacy, OUR: Our method, LAP: Laplace).

hybrid, as shown in Fig. 7 (default: hybrid). We use 10:00-19:00 in workdays as the experimental period. Every 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: $\ln(4)$).

Fig. 8 shows the evaluation results of ATD. In general, the D4D results are similar to the simulation results, and our method can always achieve a smaller ATD than Laplace. Note that among the three distribution settings, our method can gain more improvements for larger target areas (i.e., scattered and hybrid). Particularly, in the scattered setting, our method outperforms Laplace by reducing 47 percent of ATD.

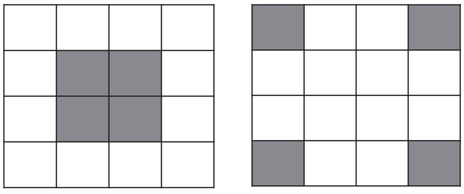


Fig. 5. Different distribution settings in simulation (a dark grid has 9× probability larger than a white grid).

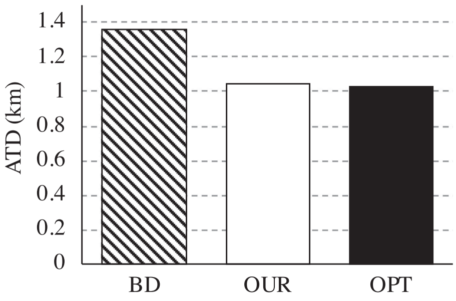


Fig. 6. ATD of our method and the optimal solution ($N_t = 2, N_c = 20, \epsilon = \ln(4)$).

In the following, we investigate the impact of 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 (Section 4.3), we measure the difference of our estimated π' and the actual π^* using Kullback-Leibler divergence [37], which can quantify how much information is lost if using π' to represent π^*

$$D_{KL}(\pi'||\pi^*) = \sum_{l\in\mathcal{L}} \pi'(l)\log\frac{\pi'(l)}{\pi^*(l)}. \tag{52}$$

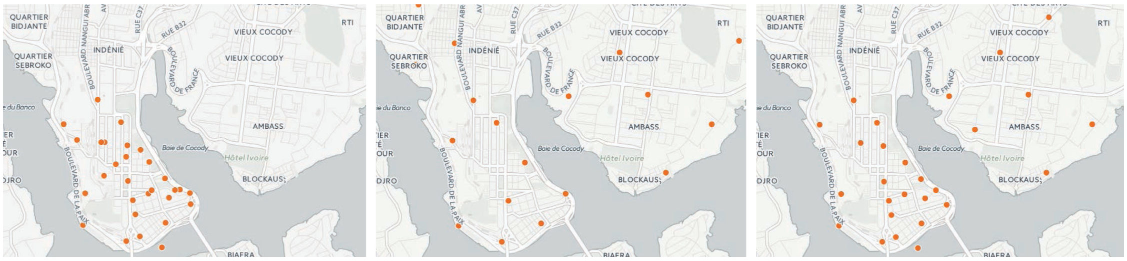
The more similar π' and π^* are, the lower D_{KL} is. Fig. 9a shows the D_{KL} for three example one-hour time slots, and we set the initial value of π' to the uniform distribution. We can see that after two or three days, D_{KL} can be reduced to about 0.2, which is much smaller than the initial D_{KL} (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 (Section 4.2), we compare it with the random selection of the initial value of $\hat{x}_0$. As shown in Fig. 9b, GA-based initialization can effectively reduce more than 10 percent of ATD to random initialization.

Runtime Performance. We use MOSEK 7.1[13] to solve our linear optimization problems. On our test PC (Intel Core i7-3612QM, 8 GB 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,

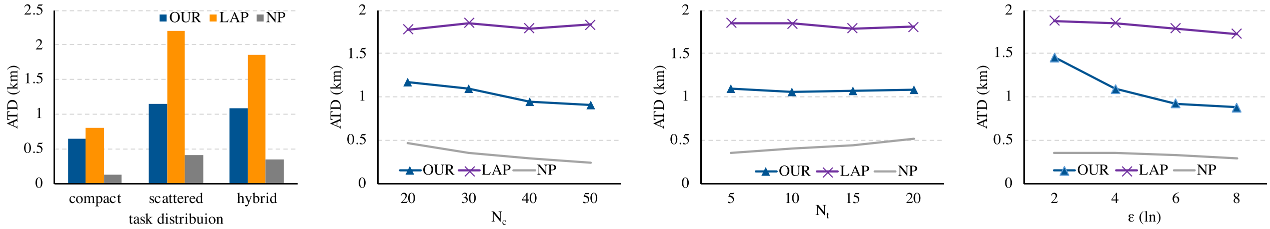
---

[13] https://www.mosek.com/



compact scattered hybrid

Fig. 7. Task distributions in D4D.



(a) vary task distribution (b) vary N_c (c) vary N_t (d) vary ϵ

Fig. 8. D4D results of average travel distance (NP: No-privacy, OUR: Our method, LAP: Laplace).

compared to no-privacy task allocation, our framework introduces an overhead of fewer than 30 seconds, which is totally acceptable in real-life MCS applications.

6.4 Balancing Differential and Distortion Privacy

In previous experiments, we set the distortion privacy level δ in our method the same as Laplace to make a fair comparison. In this section, we try to vary the setting of δ to see how it will impact the performance of our method.

It is worth noting that δ cannot be an arbitrary value since it represents the adversaries' expected inference error. A too large δ will make our optimization problem infeasible. For example, assigning δ to 100 *km* when the target area is only 10 *km* $\times$ 10 *km* is apparently not a valid setting. In practice, we can get the maximum possible δ, called δ_{max} by the following linear program:

$$\max \delta \;\; s.t. \; Eq. \; (26) \; to \; (30). \tag{53}$$

By solving this optimization problem, we find that in the D4D hybrid, δ_{max} is 1.54 *km*.

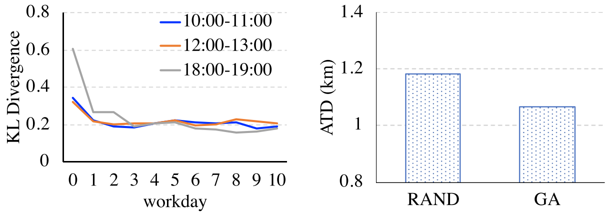
Besides, we can ignore the distortion privacy constraint (11) & (12) and solve the geo-obfuscation optimization problem (9) to get a P with *only* the ϵ-differential-privacy constraint, and then see what is the actual distortion privacy level δ that the P achieves, denoted as δ_ϵ. This can be seen as a practical lower bound of the δ configuration given ϵ, because setting δ to any value $< \delta_\epsilon$ will generate the same P as we set $\delta = \delta_\epsilon$.

Fig. 10a plots how δ_ϵ changes with ϵ. With the increase of ϵ, δ_ϵ decreases. This suggests that when we do not explicitly consider the distortion privacy in the geo-obfuscation optimization, the actually achieved distortion privacy level δ_ϵ decreases when we loosen the differential privacy level ϵ.

Given ϵ, tuning δ between δ_ϵ and δ_{max} makes a trade-off between the utility (i.e., ATD) and distortion privacy protection. We then compare ATD under different δ while fixing ϵ to $\ln(4)$ in the D4D hybrid setting (Fig. 10b). With the increase of δ, i.e., stronger distortion privacy, the utility of the obtained geo-obfuscation function decreases, i.e., ATD goes up. It is also worth noting that, the utility decreasing speed is not steady. When δ is closer to δ_{max}, increasing δ leads to higher utility loss. This observation implies that, in practice, we can sometimes obtain a certain level of distortion privacy with only a little utility loss, e.g., when we want to ensure $\ln(4)$-differential-privacy on D4D hybrid, we can increase δ to 1.4 km from $\delta_\epsilon = 1.1$ km with little utility loss; however, when we want to offer a stronger distortion privacy with $\delta > 1.4$ km, the utility loss becomes much more obvious.

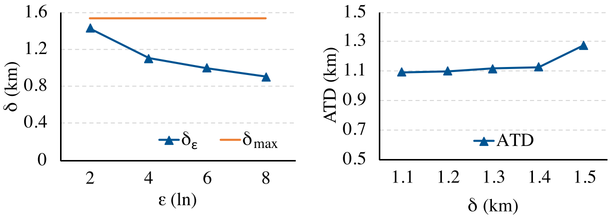
6.5 Results on Approach Extensions

Here, we evaluate our approach extensions of Section 5 with the same simulation setting as Section 6.2.



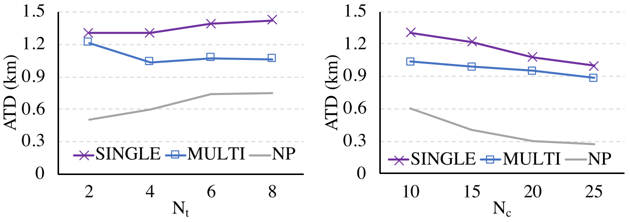
(a) Geo-distribution estimation (b) Genetic algorithm

Fig. 9. Submodule evaluation on D4D.



(a) δ_ϵ and δ_{max} (b) ATD by δ

Fig. 10. Balance differential and distortion privacy.



(a) vary N_t (b) vary N_c

Fig. 11. Results of multi-task allocation.

6.5.1 *Multi-Task Allocation*

We conduct the experiment of multi-task allocation on the simulation scenario. The target is to check how ATD changes by introducing multi-task allocation into our approach. Fig. 11 shows the ATD results of the multi-task allocation (MULTI) compared to our original single-task allocation (SINGLE), as well as No-Privacy (NP).

Fig. 11a illustrates that the improvement of MULTI over SINGLE becomes more significant with the increase of the number of tasks. Specifically, when $N_t = 8$, MULTI reduces ATD by 25.2 percent compared to SINGLE. The reason is that, when the task number increases, MULTI has more opportunities to find beneficial multi-task allocation strategies for reducing ATD. Fig. 11b indicates that when the number of candidate workers increases, the improvement of MULTI turns to be minor. The possible reason is when the candidate worker number is large, even each worker takes only one task, there are many candidate allocation strategies, and thus SINGLE can find a relatively efficient strategy. In summary, if we have fewer candidate workers and more tasks, allowing multi-task allocation will reduce ATD more significantly; and vice-versa.

6.5.2 *Multi-Objective Optimization*

In multi-objective optimization, we use acceptance ratio (AR) along with travel distance in optimization (Section 5.2). Two task acceptance models [31] are evaluated: (1) a *linear* model where users' acceptance probability drops from one when they are co-located with the task, to zero when they are 13.6 km or more away from the task (threshold is set according to a real user's crowdsourcing dataset [31]); and (2) a *power law* distribution model as follows:

$$\mathcal{A}(l, l_t) = (1 + d(l, l_t))^{-\alpha}, \tag{54}$$

where α is set to 0.5. Same as [31], we suppose that all the workers are homogeneous. Fig. 12 shows the acceptance probabilities of the two models. The β in the objective (49) for the trade-off of AR and ATD is set to 0.5 by default.

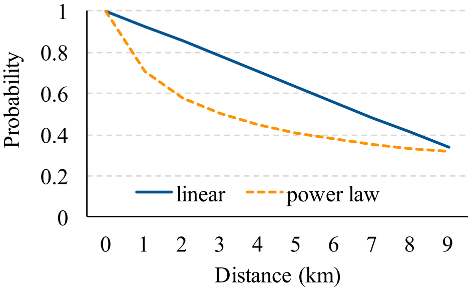


Fig. 12. Acceptance probability of two models.

The results of AR (linear model) on our simulation is shown in Fig. 13. Briefly, the results are very similar to ATD, i.e., our method can outperform Laplace significantly in all the settings. We then change the task acceptance model from the linear to the power law. Note that the power law model has a much sharper decrease in users' acceptance probability with the increase of distance (Fig. 12), and thus the AR obtaned by the power law model is lower than the linear model. In our simulation, we find that in the default setting, the power law model obtains an AR of 69.6 percent, while the linear model obtains an AR of 89.2 percent.

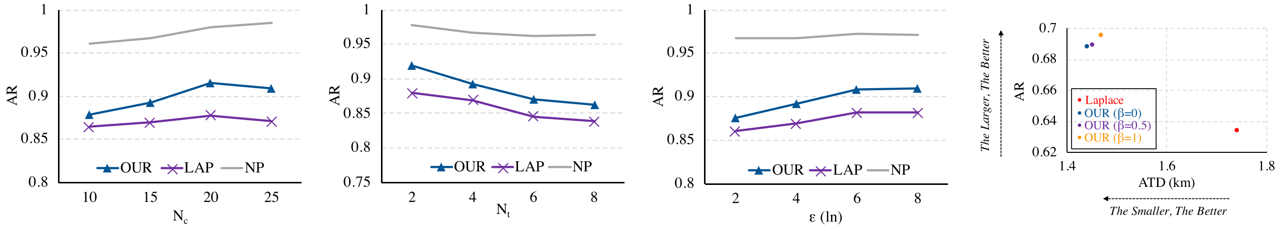
According to the Theorem 1 in Section 3.2, the setting of β does not affect our optimization results for the linear acceptance model in the simulation, but it indicates a trade-off between ATD and AR for the power law model. Hence, we show the results of ATD and AR with the power law model by varying β in Fig. 13d. We can see that, if we prioritize minimizing ATD, a smaller β can be set; if we prefer maximizing AR, a larger β can be configured. More importantly, our method consistently outperforms Laplace in both ATD and AR with any β. Our results also indicate that in practice, even if we are uncertain about the exact task acceptance model, directly minimizing travel distance usually leads to a high acceptance ratio.

7 RELATED WORK

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

7.1 Task Allocation

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* [38], [39], [40], which tries to



(a) vary N_c, linear model (b) vary N_t, linear model (c) vary ϵ, linear model (d) vary β, power law model

Fig. 13. Simulation results of acceptance ratio (NP: No-privacy, OUR: Our method, LAP: Laplace).

maximize the data quality measured by a certain metric (mostly used in environmental monitoring tasks); 2) *incentive cost* [35], [41], which aims at minimizing the total budget (from the perspective of the task organizer) for an MCS task with different incentive mechanisms, such as pay per participant [35] or pay per task [41]; 3) *energy consumption* [24], [42], whose objective is to identify an optimal collaborative data sensing and uploading scheme with energy-saving techniques such as piggybacking [24]; 4) *travel distance* [6], [27], [36], where the travel distance of a user for accomplishing a task is considered in task allocation, in order to reduce the overall travel distance for all the tasks. In this study, we emphasize on the utility on travel distance, i.e., minimizing the overall travel distance and maximizing the task acceptance ratio, as it is a critical issue for both participants and task organizers. The other kinds of utility metrics, such as the monetary budget under certain incentive mechanisms, will be studied in the future.

Multi-task allocation, as discussed in Section 5.1, has also been studied in the literature. Some studies focused on the *Worker Selected Task* model [28], [29], which is different from our studied *Server Assigned Task* model (Section 2.1). Among the related works on the SAT model, Liu et al. [27] considered multi-task allocation in emergency scenarios (e.g., heavy rain) when the total task number is larger than the worker capacity; however, in our case, the task number is smaller than the worker number, and thus the method in [27] cannot be applied. Deng et al. [30] considered a strict constraint in multi-task allocation—each task has its *hard deadline* and an assigned worker should complete the task before it. With geo-obfuscation, it is hard to estimate the time when a worker can finish a task due to location uncertainty; so designing an assignment strategy to satisfy the hard deadline constraint becomes more challenging. We leave this to our future work.

7.2 Location Privacy

Location privacy in MCS has attracted increasing research interests. Based on a recent survey on the MCS privacy issues [8], cloaking is still a widely used strategy in practice for protecting location privacy, e.g., [9], [10], [11]. 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. [43] proposed to leverage differential geo-obfuscation in environmental 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 environmental monitoring tasks. A closely related work to ours is presented in [7], which also attempted to optimize workers' travel distance under differential privacy protection. However, their mechanism needs a third-party trusted entity to first collect users' real locations before perturbation. They proposed to let users' cellular service providers act like such a third-party, but incentivizing service providers for participation is hard in practice. In our solution, mobile users obfuscate their locations on their smartphones, thus avoiding such a trusted third-party.

Oya et al. [44] studied differential geo-obfuscation on the inference error of confidence, i.e., the probability that an adversary chooses a wrong location given two candidates (one correct and one wrong). We note that this inference error [44] is inherently equivalent to differential privacy and distinct from the inference error of distortion privacy (measured by distance) in our work. Recently, the original differential geo-obfuscation definition (1) has been redefined for certain practical cases. For example, Chatzikokolakis et al. [45] proposed an alternative definition to modify the distance metric d in the definition (1) from the euclidean distance to an elastic metric which considers the location characteristics (space, population, points-of-interests, etc.). Our method can be easily extended to such alternative differential geo-obfuscation definitions by simply replacing the original definition (1) with the new one such as [45].

Compared to our preliminary conference version [46], this journal paper has improved its substantial technical part from three perspectives: (1) adding *distortion privacy* in company with differential privacy; (2) extending the approach toward *multi-task allocation*; (3) realizing *multi-objective optimization* considering both travel distance and task acceptance ratio.

8 DISCUSSION AND FUTURE WORK

More Tasks & Fewer Workers. Our paper assumes more workers & fewer tasks (abbr. MWFT, Section 2.1), while the opposite case (more tasks & fewer workers, abbr. MTFW) might also happen. For instance, in emergency scenarios like heavy rain, there could be fewer workers while more tasks (e.g., check traffic condition) on the roads [27]. Our solution can deal with MTFW with minor modifications. Specifically, to maximize the number of assigned tasks in MTFW, every worker will be assigned with one task (if not considering multi-task allocation). Hence, when formulating MINLP (9) in Section 3, we modify '$\leq$' to '$=$' in Eq. (16) as one user can only conduct one task.[14] Besides, we change '$=$' to '$\leq$' in Eq. (13) as not all the tasks can be assigned in MTFW. For this modified MINLP, our solution in Section 4 can still work. If allowing multi-task allocation in MTFW, we can also refer to Section 5.1 for the solution extension.

Theoretical Performance Analysis. In this work, we propose a method combing BD and GA for solving MINLP (9). As BD and GA are heuristic methods, the theoretical performance guarantee of the solution is unclear. While we have empirically validated the performance of our solution by comparing it to the global optima under a small-scale simulation test, we will try to conduct theoretical analysis in the future.

Trajectory Protection. Our current research provides the *snapshot* protection. In reality, an MCS worker may conduct multiple tasks continuously and then reveal her/his moving trajectory. A simple adaptation of the snapshot protection to the trajectory protection with differential geo-obfuscation is setting smaller ϵ [12], but this will lead to significant utility loss in practice. Hence, how to protect trajectory privacy is still an open question to investigate.

Task Location Protection. As task locations may reveal task owners' sensitive information [47], some recent studies started considering task location protection with techniques such as Laplace differential geo-obfuscation [48]. In the future, we will study whether optimized differential-and-distortion privacy protection can be incorporated to protect both worker and task locations.

---

[14] Rounding (floor or ceiling) may be needed as it is a integer program.

9 CONCLUSION

This paper addresses the privacy-preserving problem in MCS task allocation. It uses differential-and-distortion geo-obfuscation to protect users' location privacy without the need to involve any trusted third-party service. Meanwhile, it aims at minimizing workers' travel distance. To this end, this paper proposes a mixed-integer nonlinear program to collectively optimize both differential-and-distortion geo-obfuscation and task allocation, using the techniques including Benders decomposition, genetic algorithms, and Bayesian analysis. The approach is further extended to multi-task allocation and multi-objective optimization.

ACKNOWLEDGMENTS

This project is partially supported by NSFC Grant nos. 61572048 and 71601106, State Language Commission of China Key Program Grant no. ZDI135-18, Hong Kong ITF Grant no. ITS/391/15FX, and ERC Consolidator Grant no. 683253 (GraphInt). We would like to thank T. Wang for preprocessing the D4D dataset used in the experiment, and H. To for providing the source codes of [31] for our reference.

REFERENCES

[1] T. Yan, M. Marzilli, R. Holmes, D. Ganesan, and M. Corner, "mCrowd: A platform for mobile crowdsourcing," in *Proc. 7th ACM Conf. Embedded Netw. Sensor Syst.*, 2009, pp. 347–348.

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

[3] D. Zhang, L. Wang, H. Xiong, and B. Guo, "4W1H in mobile crowd sensing," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 42–48, Aug. 2014.

[4] L. Chen and C. Shahabi, "Spatial crowdsourcing: Challenges and opportunities," *IEEE Data Eng. Bulletin*, vol. 39, no. 4, pp. 14–25, Dec. 2016.

[5] B. Guo, Z. Wang, Z. Yu, Y. Wang, N. Y. Yen, R. Huang, and X. Zhou, "Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm," *ACM Comput. Surveys*, vol. 48, no. 1, 2015, Art. no. 7.

[6] B. Guo, Y. Liu, W. Wu, Z. Yu, and Q. Han, "ActiveCrowd: A framework for optimized multitask allocation in mobile crowdsensing systems," *IEEE Trans. Human-Mach. Syst.*, vol. 47, no. 3, pp. 392–403, Jun. 2017.

[7] H. To, G. Ghinita, and C. Shahabi, "A framework for protecting worker location privacy in spatial crowdsourcing," *Proc. VLDB Endowment*, vol. 7, no. 10, pp. 919–930, 2014.

[8] 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 Rec.*, vol. 44, no. 4, pp. 23–34, 2016.

[9] C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos, "Anonysense: Privacy-aware people-centric sensing," in *Proc. Int. Conf. Mobile Syst. Appl. Serv.*, 2008, pp. 211–224.

[10] L. Pournajaf, L. Xiong, V. Sunderam, and S. Goryczka, "Spatial task assignment for crowd sensing with cloaked locations," in *Proc. Int. Conf. Mobile Data Manage.*, 2014, pp. 73–82.

[11] I. J. Vergara-Laurens, D. Mendez, and M. A. Labrador, "Privacy, quality of information, and energy consumption in participatory sensing systems," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, 2014, pp. 199–207.

[12] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 901–914.

[13] C. Dwork, "Differential privacy: A survey of results," in *Proc. Int. Conf. Theory Appl. Models Comput.*, 2008, pp. 1–19.

[14] N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Optimal geo-indistinguishable mechanisms for location privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2014, pp. 251–262.

[15] R. Shokri, "Privacy games: Optimal user-centric data obfuscation," *Proc. Privacy Enhancing Technol.*, vol. 2015, no. 2, pp. 1–17, 2015.

[16] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.-Y. Le Boudec , "Protecting location privacy: Optimal strategy against localization attacks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2012, pp. 617–627.

[17] J. F. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Numerische Mathematik*, vol. 4, no. 1, pp. 238–252, 1962.

[18] L. Willenborg and T. De Waal, *Elements of Statistical Disclosure Control*. Berlin, Germany: Springer, 2012.

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

[20] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[21] A. H. Land and A. G. Doig, "An automatic method of solving discrete programming problems," *Econometrica: J. Econometric Soc.*, vol. 28, pp. 497–520, 1960.

[22] A. M. Geoffrion, "Generalized benders decomposition," *J. Optimization Theory Appl.*, vol. 10, no. 4, pp. 237–260, 1972.

[23] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1998.

[24] H. Xiong, D. Zhang, L. Wang, and H. Chaouchi, "EMC 3: Energy-efficient data transfer in mobile crowdsensing under full coverage constraint," *IEEE Trans. Mobile Comput.*, vol. 14, no. 7, pp. 1355–1368, Jul. 2015.

[25] K.-R. Koch, *Introduction to Bayesian Statistics*. Berlin, Germany: Springer, 2007.

[26] A. K. Dey, "Understanding and using context," *Pers. Ubiquitous Comput.*, vol. 5, no. 1, pp. 4–7, 2001.

[27] Y. Liu, B. Guo, Y. Wang, W. Wu, Z. Yu, and D. Zhang, "TaskMe: Multi-task allocation in mobile crowd sensing," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2016, pp. 403–414.

[28] Y. Li, M. L. Yiu, and W. Xu, "Oriented online route recommendation for spatial crowdsourcing task workers," in *Proc. Int. Symp. Spatial Temporal Databases*, 2015, pp. 137–156.

[29] D. Deng, C. Shahabi, and U. Demiryurek, "Maximizing the number of worker's self-selected tasks in spatial crowdsourcing," in *Proc. 21st ACM SIGSPATIAL Int. Conf. Advances Geographic Inf. Syst.*, 2013, pp. 324–333.

[30] D. Deng, C. Shahabi, and L. Zhu, "Task matching and scheduling for multiple workers in spatial crowdsourcing," in *Proc. 23rd SIGSPATIAL Int. Conf. Advances Geographic Inf. Syst.*, 2015, Art. no. 21.

[31] H. To, G. Ghinita, L. Fan, and C. Shahabi, "Differentially private location protection for worker datasets in spatial crowdsourcing," *IEEE Trans. Mobile Comput.*, vol. 16, no. 4, pp. 934–949, Apr. 2017.

[32] M. Musthag and D. Ganesan, "Labor dynamics in a mobile micro-task market," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2013, pp. 641–650.

[33] C.-L. Hwang and A. S. M. Masud, *Multiple Objective Decision Making - Methods and Applications: A State-of-the-Art Survey*. Berlin, Germany: Springer, 2012.

[34] 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:1210.0137, Sep. 2012.

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

[36] S. He, D.-H. Shin, J. Zhang, and J. Chen, "Toward optimal allocation of location dependent tasks in crowdsensing," in *Proc. IEEE INFOCOM*, 2014, pp. 745–753.

[37] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.

[38] L. Wang, D. Zhang, Y. Wang, C. Chen, X. Han, and A. M'hamed, "Sparse mobile crowdsensing: Challenges and opportunities," *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 161–167, Jul. 2016.

[39] 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 *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2015, pp. 683–694.

[40] Y. Zhu, Z. Li, H. Zhu, M. Li, and Q. Zhang, "A compressive sensing approach to urban traffic estimation with probe vehicles," *IEEE Trans. Mobile Comput.*, vol. 12, no. 11, pp. 2289–2302, Nov. 2013.

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

[42] X. Sheng, J. Tang, and W. Zhang, "Energy-efficient collaborative sensing with mobile phones," in *Proc. IEEE INFOCOM*, 2012, pp. 1916–1924.

[43] L. Wang, D. Zhang, D. Yang, B. Y. Lim, and X. Ma, "Differential location privacy for sparse mobile crowdsensing," in *Proc. IEEE Int. Conf. Data Mining*, 2016, pp. 1257–1262.

[44] S. Oya, C. Troncoso, and F. Pérez-González, "Is geo-indistinguishability what you are looking for?" in *Proc. Workshop Privacy Electron. Soc.*, 2017, pp. 137–140.

[45] K. Chatzikokolakis, C. Palamidessi, and M. Stronati, "Constructing elastic distinguishability metrics for location privacy," *Proc. Privacy Enhancing Technol.*, vol. 2015, no. 2, pp. 156–170, 2015.

[46] L. Wang, D. Yang, X. Han, T. Wang, D. Zhang, and X. Ma, "Location privacy-preserving task allocation for mobile crowdsensing with differential geo-obfuscation," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 627–636.

[47] A. Liu, W. Wang, S. Shang, Q. Li, and X. Zhang, "Efficient task assignment in spatial crowdsourcing with worker and task privacy protection," *GeoInformatica*, vol. 22, pp. 335–362, 2018.

[48] H. To, C. Shahabi, and L. Xiong, "Privacy-preserving online task assignment in spatial crowdsourcing with untrusted server," in *Proc. IEEE 34th Int. Conf. Data Eng.*, 2018, pp. 833–844.

Leye Wang received the PhD degree in computer science from the Institut Telecom SudParis, University Paris 6, France, in 2016, and he was a postdoc researcher with the Hong Kong University of Science and Technology. He is an assistant professor with the Key Lab of High Confidence Software Technologies, Peking University, China. His research interests include ubiquitous computing, mobile crowdsensing, and urban computing.

Dingqi Yang received the PhD degree in computer science from Pierre and Marie Curie University and Institut Mines-TELECOM/TELECOM SudParis, where he won both the CNRS SAMOVAR Doctorate Award and the Institut Mines-TELECOM Press Mention, in 2015. He is currently a senior researcher with the University of Fribourg in Switzerland. His research interests include big social media data analytics, ubiquitous computing, and smart city applications.

Xiao Han received the PhD degree in computer science from Pierre and Marie Curie University and Institut Mines-TELECOM/TELECOM SudParis, in 2015. She is currently an assistant professor with the Shanghai University of Finance and Economics in China. Her research interests include social network analysis, fintech, and privacy protection.

Daqing Zhang (IEEE Fellow) received the PhD degree from the University of Rome La Sapienza, in 1996. He is a professor with the School of EECS, Peking University, China and Telecom SudParis, France. His research interests include context-aware computing, urban computing, mobile computing, big data analytics, pervasive elderly care, etc.. He has published more than 260 technical papers in leading conferences and journals. He served as the general or program chair for more than 10 international conferences, giving keynote talks at more than 20 international conferences. He is the associate editor of the *ACM Transactions on Intelligent Systems and Technology*, the *IEEE Pervasive Computing*, etc.. He is the winner of the Ten-years CoMoRea impact paper award at IEEE PerCom 2013, the Honorable Mention Award at ACM UbiComp 2015 and 2016, the Best Paper award at IEEE UIC 2015 and 2012. He is a fellow of the IEEE.

Xiaojuan Ma received the PhD degree in computer science from Princeton University. She is an assistant professor of Human-Computer Interaction (HCI), Department of Computer Science and Engineering (CSE), Hong Kong University of Science and Technology (HKUST). She was a post-doctoral researcher with the Human-Computer Interaction Institute (HCII), Carnegie Mellon University (CMU), and before that a research fellow with the Information Systems Department, National University of Singapore (NUS). Before joining HKUST, she was a researcher of Human-Computer Interaction, Noah's Ark Lab, Huawei Tech. Investment Co., Ltd. in Hong Kong. Her background is in Human-Computer Interaction. She is particularly interested in data-driven human-engaged computing in the domain of ubiquitous, social, and crowd computing and Human-Robot Interaction. She was the recipient of Computing Innovation Fellows by Computing Research Association in 2010, and named Outstanding Chinese Young Leaders in HCI by the International Chinese Association of Computer Human Interaction in 2016.