Task Allocation under Geo-Indistinguishability via Group-based Noise Addition

Pengfei Zhang, Xiang Cheng, Sen Su, and Ning Wang

Abstract—Locations are usually necessary for task allocation in spatial crowdsourcing, which may put individual privacy in jeopardy without proper protection. Although existing studies have well explored the problem of location privacy protection in task allocation under geo-indistinguishability, they potentially assume the workers could perform any tasks, which might not be practical in reality. Moreover, they usually adopt planar laplacian mechanism to achieve geo-indistinguishability, which will introduce excessive noise due to its randomness and boundlessness. To this end, we propose a task allo**CA**tio**N** approach via gr**O**up-based nois**E** addition under Geo-I, referred to as *CANOE*. Its main idea is that each worker uploads the noisy distances between his true location and the obfuscated locations of his preferred tasks instead of uploading his obfuscated location. In particular, to alleviate the total noise when conducting grouping, we put forward an optimized global grouping with adaptive local adjustment method *OGAL* with convergence guarantee. To collect the noisy distances which are required for subsequent task allocation, we develop a utility-aware obfuscated distance collection method *UODC* with solid privacy and utility guarantees. We further theoretically analyze the privacy, utility and complexity guarantees of *CANOE*. Extensive analyses and experiments over two real-world datasets confirm the effectiveness of *CANOE*.

Index Terms—Spatial Crowdsourcing, Task Allocation, Privacy Protection, Geo-indistinguishability.

1 Introduction

Spatial crowdsourcing (SC) is a typical representative of crowdsourcing, which engages the crowds to accomplish spatial tasks, e.g., giving out coupons and monitoring environmental conditions, by physically moving to other locations [1]. In SC, due to the limitation of time or space, each worker can only be assigned to do the limited tasks initiated by the initiator. For example, an individual worker may only have time to do nearby tasks that are near the places of work or residence. The task that is far away from these locations should not be assigned to him. Thus, how to assign a certain worker the task that he can do objectively is a key issue. Task allocation [2], [3], [4], which assigns a worker to his preferred task under the constraint on minimizing average travel distance, is an effective way to address this issue. In particular, average travel distance denotes the average distance needed for the selected workers to perform their assigned tasks. The smaller, the better. For instance, in location-based services, such as personalized product recommendation [5], [6], the initiators of the spatial crowdsourcing want different workers to be responsible for product recommendation in different regions. Obviously, since each worker has different familiarities with different regions, to maximize the quality of recommendations, the problem can be modeled as the task allocation problem while considering preferences. However, disclosing the naturally required locations may pose a significant risk to the workers as the uploaded locations might imply living habits or other sensitive information. Without proper privacy protection, the workers may be unwilling to participate in the SC system.

Geo-indistinguishability (Geo-I) [7] has been proposed as a practical standard to address location privacy protection problem. Unlike the spatial cloaking technique [8], which replaces a worker's location with a restricted region, the protection level of Geo-I is independent of adversaries' prior knowledge. Moreover, it does not assume a trusted server. Furthermore, it can be achieved in a simple and efficient way using the planar laplacian (*PL*) mechanism. Through *PL*, locations will be locally perturbed before being sent to the server. Since exact locations never leave from workers' devices, it can protect both workers and other adversaries, such as the server, against damages due to potential privacy breaches. Recently, *PL* has been adopted in many famous applications, including LP-Guardian [9], LP-Doctor [10] and secure nearby-friends discovery system [11].

In this paper, we systematically study the problem of task allocation with preferred tasks while protecting workers' locations under Geo-I. Specifically, each worker first excludes the tasks he doesn't want to do and lets the remaining tasks as his preferred tasks or just chooses the tasks he wants to do. Then the server assigns each task to a worker under the constraint on minimizing average travel distance while guaranteeing Geo-I for each worker's location. Although existing studies [12], [13], [14], [15], [16], [17] have well explored the problem of location privacy protection for task allocation, they potentially assume the workers could perform any tasks, which might not be practical in reality. Moreover, due to the randomness and boundlessness of *PL* [18], the injected noise may be excessive.

We note that in the real world, workers prefer to select nearby tasks that are near the places of work or residence. It leads to the aggregations of tasks' locations in the corresponding preference sets. Thus, if some tasks are close to each other in terms of locations, they can be represented by one location. This fact drives us to advocate the idea of group-based noise addition. It is intended to strike a balance between location privacy and location utility. Specifically, as an extreme example, when multiple locations in a certain worker's preference set are the same, there is even no extra information loss except for the added noise in the way of adding noise by using a location in this preference set, and the noise needs to be added only once. To improve the utility of task allocation under Geo-I, we propose to add noise by group-based noise

- *Pengfei Zhang, Xiang Cheng, Sen Su and Ning Wang are with State Key Laboratory of Networking and Switching Tech., Beijing University of Posts ands Telecomm. (BUPT), China.*
  (E-mail: {zpf.bupt, chengxiang, susen, wangn}@bupt.edu.cn)
- *Corresponding author of this paper is Prof. Xiang Cheng and Prof. Sen Su.*

Manuscript received XX,XX 2021.

addition on the basis of minimizing the overall noise. Note that, in this way, we may need to split the privacy budget for multiple grouping representative locations. Although directly announcing the obfuscated workers' locations with *PL* guarantees Geo-I and avoids the problem of splitting the privacy budget, the obfuscated locations might deviate from the true locations significantly, as the injected noise of adopting *PL* for achieving Geo-I is unbounded, which will lead to excessive noise of the calculated distances. In contrast, in the way of group-based noise addition, since each generated obfuscated location is limited to the range formed by the corresponding tasks' locations in the same group, desirable data utility can be expected. Moreover, the privacy budget of each worker only needs to be split just a few times in practice due to the existence of the aggregations of the tasks' locations in each preference set. As such, we believe that *CANOE* can provide an acceptable trade-off for real-world deployments.

To this end, we propose a task allo**CA**tio**N** approach via gr**O**up-based nois**E** addition under Geo-I, referred to as *CANOE*. Its main idea is that each worker uploads the noisy distances between his true location and the obfuscated locations of his preferred tasks instead of uploading his obfuscated location. In particular, we use one location to represent a group of geographically close locations. This indicates only one piece of noise will be added to the above locations in the same group.

In *CANOE*, two kinds of noise are involved in obtaining the appropriate groupings, which are the information loss caused by representing multiple locations by one location and the injected noise for privacy protection. We group the locations in each worker's preference set through the constraints deduced by minimizing the sum of the above two types of noise, and put forward an **o**ptimized global **g**rouping with **a**daptive **l**ocal adjustment method called *OGAL* based on our noise analysis. In *OGAL*, the server first gives coarse-grained global groupings and broadcasts the groupings to each worker. Then, each worker refines the received groupings locally. Specifically, to conduct grouping globally, we formulate a mixed-integer nonlinear program (*MINLP*) problem with a non-convex constraint. Due to the NP-hard characteristic, based on benders decomposition (*BD*) and alternating direction method of multipliers (*ADMM*) [19], we devise an alternate optimization method with convergence guarantee. To refine groupings locally, we formalize two optimization problems to adaptively determine the grouping for each worker, including the number of groups and the number of tasks in each group.

After obtaining the optimized grouping for each worker, he adds noise for each group to collect and uploads the required distances for subsequent task allocation. We note that we should make the generated representative location for each group be close to tasks' locations in the same group and be far away from tasks' locations in other groups. In addition, recall that the straight-forward adoption of *PL* may introduce excessive noise. To this end, instead of adopting *PL*, we develop a utility-aware obfuscated distance calculation method called *UODC*. Its main idea is to let each generated obfuscated location be limited to the range formed by tasks' locations in the same group. We theoretically prove it guarantees Geo-I and show its utility superiority compared with *PL*. In *UODC*, we first formalize an optimization problem to generate a grouping representative location for each group by minimizing intra-group distances and maximizing inter-group distances. Then, we generate a noisy representative location by probability sampling with each corresponding group. Finally, each worker uploads the calculated noisy distances from his true location to each noisy representative location to the server.

The key contributions of this paper are summarized as follows:

- We present *CANOE* for task allocation under Geo-I based on group-based noise addition. Its main idea is that each worker uploads the noisy distances between his true location and the obfuscated locations of his preferred tasks instead of uploading his obfuscated location.
- To alleviate the total noise when conducting grouping, we put forward an optimized global grouping with adaptive local adjustment method *OGAL* based on our noise analysis. We prove its convergence guarantee.
- To collect the noisy distances which are required for subsequent task allocation, we develop a utility-aware obfuscated distance collection method *UODC*. We formally give its privacy and utility guarantees.
- We theoretically give the privacy, utility and complexity guarantees of *CANOE*. Extensive analyses and experiments over two real-world datasets confirm the effectiveness of *CANOE*.

The rest of this paper is organized as follows. We discuss related work in Section 2. We give the preliminaries in Section 3. The details of *CANOE* are presented in Section 4. The experimental results are discussed and analyzed in Section 8. Finally, we summarize our work in Section 9.

2 Related Work

The related work falls into the following two aspects.

2.1 Privacy-preserving Task Allocation with Obfuscation

We briefly discuss existing privacy-preserving task allocation approaches in spatial crowdsourcing while adopting obfuscation technologies (e.g., geo-indistinguishability (Geo-I) [7] or differential privacy [20]).

Zhang et al. [21] conduct task allocation based on a designed differentially private geocoding method to preserve workers' locations. To et al. [12] propose a differentially private framework for protecting the privacy of workers' locations. They [13] further propose an analytical method to ensure the high success rate of task allocation. To improve task allocation utility, Wang et al. [14] integrate the phase of adding noise and the phase of task allocation into an optimization problem together while guaranteeing differential privacy. They [15] also propose a method to maximize each mobile worker's future location coverage under a guaranteed location privacy protection scheme. Similar to [15], Qian et al. [22] focus on improving the service quality using task allocation for vehicle networks. In addition, they [23] design a similar solution for mobile edge cloud environment. To satisfy personalized demand for each worker, Wang et al. [24] provide a personalized probabilistic winner selection mechanism. However, the utility of [24] significantly depends on the choice of hyper parameters, which could not be automatically determined. For jointly protecting the location privacy of workers and tasks, To et al. [16] propose a three-phase framework which quantifies the achievable probabilities. Following [16], Tao et al. [17] make the first attempt at differentially private task allocation while assuring the competitive ratio. However, these two mainly [16], [17] focus on task allocation aiming to ensure all tasks are assigned, while we focus on minimize the expected travel distance of selected workers.

Task allocation with obfuscation technologies has also been studied whiling incorporating incentive. Shen et al. [25] aim to optimize task acceptance rate using a leader-follower game by introducing edge nodes. Gong et al. [26] propose a framework

to achieve high task coverage by estimating worker density. Xu et al. [27] propose a differential privacy-based auction mechanism in cloud and edge-cooperation systems. Zhang et al. [28] design a game-theoretic-based task allocation approach for social sensing-based edge computing systems. However, since they work on the problems that are different from ours, their approaches are not suitable for our problem.

In summary, although existing studies can prevent location privacy leakage, they potentially assume the workers could perform any tasks. Moreover, some focus on different objectives or different scenarios from us. Furthermore, due to the randomness and boundlessness of *PL*, the obfuscated locations of the workers might deviate from the true locations significantly. In this paper, we focus on the scenario when workers have preferred tasks while designing a new noise-adding mechanism to collect the information required for task allocation instead of adopting *PL*. In particular, we present group-based noise addition to add noise for the preferred tasks' locations rather than workers' locations, which can provide desirable utility while keeping the same intensity of privacy protection.

2.2 Location Privacy under Geo-indistinguishability

Large amounts of work have been done to protect the location privacy, and many excellent surveys, such as, [29] and [30], have conducted systematic studies on the classic methods of location privacy protection. In particular, the notion of Geo-I [7] has gained wide attention.

Bordenabe et al. [31] explore the possibility of constructing a mechanism that minimizes the service quality loss, using linear programming technique. Yu et al. [32] propose another notion to supplement Geo-I. Oya et al. [33] present a new method to maximize the conditional entropy utility. Pyrgelis et al. [34] focus on releasing aggregated location time-series. Chatzikokolakis et al. [35] aim to improve utility for continuous and discrete scenarios. ElSalamouny et al. [36] consider a more realistic case in which the region to be protected is continuous with a non-zero area. Takagi et al. [37] identify an extra privacy loss for road networks. Oya et al. [38] provide an alternative formulation of Geo-I.

Though existing studies may provide good utility for a single location, they are no longer capable of obtaining reliable results under our setting. On one hand, the obfuscated locations could be far from the true locations. On the other hand, we cannot directly use *PL* for multiple locations as the privacy budget could be very small, which will inevitably result in poor performance. Besides, the performances of [33], [34] rely heavily on the choice of hyper parameters, which may result in the different results.

3 Preliminaries

3.1 Geo-indistinguishability

Geo-indistinguishability (Geo-I) is the *de facto* for protecting location privacy [7]. Let $\mathcal{X}$ and $\mathcal{Z}$ be the set of workers' possible locations and possible obfuscated locations respectively. It is formally defined as follows:

Definition 1 (ε-Geo-I)**.** *Given a privacy budget ε, a randomized algorithm achieves ε-Geo-I, if and only if*

$$P\left(l^* \left| l_1 \right.\right) \leq e^{\varepsilon r} P\left(l^* \left| l_2 \right.\right),$$

where l_1 and l_2 are any two locations in $\mathcal{X}$, and l^* is an observed obfuscated location in $\mathcal{Z}$. In particular, r denotes the distance from true location to the observed obfuscated location.

The way to guarantee Geo-I is adopting the planar laplacian (*PL*). Specifically, given the privacy budget parameter ε, the actual location $l_0 \in \mathcal{X}$, and any other location $l \in \mathcal{Z}$, we can generate a noisy location by:

$$l = l_0 + \left(r * \cos\left(\theta\right), r * \sin\left(\theta\right)\right).$$

In particular, θ is sampled from $[0, 2\pi)$, and r is obtained by:

$$r = -\frac{1}{\varepsilon}\left(W_{-1}\left(\frac{p-1}{e}\right) + 1\right),$$

where W_{-1} is the Lambert function (the -1 branch) and $p \in [0, 1)$.

To support multiple computations under Geo-I, composite properties are extensively used, including sequential composition property [20] and parallel composition property [39]. In particular, we have:

Theorem 1. *(Sequential Composition) Suppose $\mathcal{A}_1, ..., \mathcal{A}_k$ are k algorithms over database D, each provides ε_i-Geo-I. A sequence of algorithms $\mathcal{A}_i(D)$ yields $(\sum \varepsilon_i)$-Geo-I.*

3.2 Problem Statement

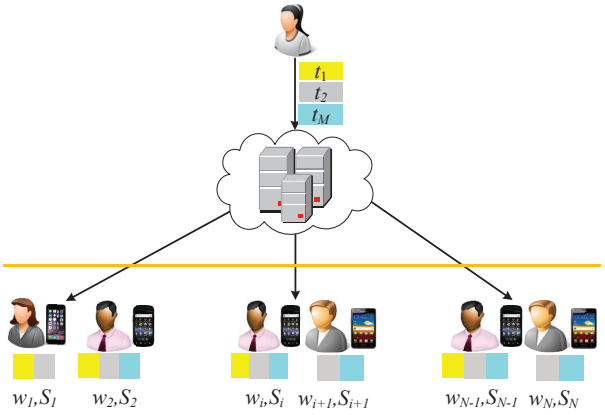


Figure 1. The research problem

Fig. 1 shows the problem investigated in this paper. Given a crowd of active workers in the SC system, denoted by the set $W = \{w_1, w_2, w_3, \cdots, w_M\}$. There exist N $(N < M)$ tasks published by the initiator to be accomplished, which is represented as $T = \{t_1, t_2, t_3, \cdots, t_N\}$. Each task has its location coordinate and corresponding identifier. Each worker has his own location coordinate and preference set S_i with the size of ρ_i, in which the elements are the identifiers of the tasks a worker prefers to do. Additionally, we use $d\left(w_i, t_j\right)$ to denote the distance the i-th worker w_i travels if he is selected to perform the j-th task t_j, which is in his preference set.

We assume that the size of the preference set for each worker is the same, denoted by ρ, and each worker uses the same privacy budget ε. In addition, suppose that the true location of each task is publicly available. Furthermore, we assume that the initiator has recruited adequate workers to evenly prefer each task through the existing *Target Coverage-based worker recruitment* approaches [40], [41], [42], [43], [44], [45], which is beyond the scope of this paper. How to jointly model worker recruitment and task allocation [46], [47], [48], under Geo-I, is an interesting issue that will be studied in the future. Once the server assigns a task to a worker, the worker and the task are considered being unreachable. Specifically, our problem is to identify a set of tuples with form $< w, t >$, where a spatial task t is assigned to worker w, satisfying that t is a task in the preference set of w. It aims to minimize traveling distance while completing all the tasks with the obfuscated information provided by the workers under ε-Geo-I.

Table 1 summarizes the notations that will be frequently used in this paper.

TABLE 1: List of Frequently Used Notations

Notation	Definition
W	The set of workers
T	The set of tasks
M	The total number of workers
N	The total number of tasks
w_i	The *i*-th worker
t_j	The *j*-th task
S_i	The preference set for w_i
ρ	The size of the preference set
$d(\cdot)$	The travel distance
$\tilde{d}(\cdot)$	The obfuscated distance
μ	The representative location for a group
μ'	The representative location after adding noise
$\widetilde{d_{ij}}$	The obfuscated distance between w_i and $\mu_j^{'}$
C	The grouping set of a certain preference set
E_G	The injected noise
E_{IL}	The information loss
$d(w_i, t_j)$	The distance travelled by w_i for performing t_j
D	The noisy distance matrix

4 Overview of CANOE

The intuitive scheme is that each worker adds noise using the planar laplacian (*PL*) to obfuscate his true location, and then calculates the distances between the corresponding obfuscated location and his preferred tasks' locations. After that, the noisy distances are sent to the server for task allocation. Nevertheless, due to the randomness and boundlessness of *PL*, it may imply excessive noise to be added.

Actually, due to the fact that workers prefer to select nearby tasks that are near the places of work or residence, there exist the aggregations of tasks' locations in each preference set. Thus, if some tasks are close to each other in terms of locations, they can be represented by one location. This fact drives us to advocate the idea of group-based noise addition. Based on this idea, we present *CANOE*, whose main idea is that each worker uploads the noisy distances between his true location and the obfuscated tasks' locations in his preference set instead of uploading his obfuscated location directly. In particular, *CANOE* mainly consists of an optimized global grouping with adaptive local adjustment method *OGAL* and a utility-aware obfuscated distance collection method *UODC*. Fig. 2 depicts the workflow of *CANOE*, in which the server and the workers collaborate to assign the tasks in each worker's preference set in a geo-indistinguishable manner. In particular, *CANOE* includes the following phases.

Phase 1. In this phase, the server first invokes *OGAL* to group all the tasks globally according to their publicly available locations by formulating a mixed-integer nonlinear program *MINLP* problem with a non-convex constraint, and then broadcasts the groupings to each worker. The details are given in Subsection 5.1.

Phase 2. In this phase, each worker first applies the groupings sent by the server to initially group the locations in his preference set, and then invokes *OGAL* to adaptively refine his grouping based on the formalized two optimization problems, including the determination of the number of groups and the number of tasks in each group. The details are discussed in detail in Subsection 5.2.

Phase 3. In this phase, after local adjustment, each worker already gets his optimal grouping, which may be different from each other. He invokes *UODC* to calculate the noisy distances that are used for task allocation, and uploads the noisy distances to the server. *UODC* is clearly elaborated in Section 6.

Phase 4. In this phase, based on the uploaded noisy distances, the server conducts task allocation to assign each task to a certain worker according to the algorithm described in [2].

Besides, we give the privacy, utility and complexity guarantees of *CANOE* in Section 7.

5 Optimized Global Grouping with Adaptive Local Adjustment

Clearly, two kinds of noise are involved in obtaining the appropriate groupings. We denote the noise for privacy protection as *Injected Noise*, and denote the reconstruction error caused by representing a set of locations by one location as *Information Loss*. Too many groups for a certain preference set will lead to excessive *Injected Noise*, while too few groups will imply too much *Information Loss*.

Thus, to strike a balance between location privacy and location utility, we put forward an **o**ptimized global **g**rouping with **a**daptive **l**ocal adjustment method *OGAL* to minimize the total noise. The overflow of *OGAL* is shown in Fig. 3. In *OGAL*, the server first gives the coarse-grained groupings and broadcasts the groupings to each worker. Then, each worker adjusts his own groupings locally.

In what follows, we first introduce the global grouping method. Then, we present how to adjust groupings locally. Before giving the details of the global grouping method, we first quantify the total noise, which is defined by the sum of the *Injected Noise* and *Information Loss*.

5.1 Optimized Global Grouping

5.1.1 Total Noise Quantification

The *Injected Noise* is the expectation of noise added for guaranteeing Geo-I.

Definition 2 (*Injected Noise*)**.** *Formally, for a certain group C_k, the noise added to the group center is*

$$\begin{aligned} E_G(C_k) &= \int_0^{+\infty}\int_0^{2\pi} rD(r,\theta)\,drd\theta \\ &= \int_0^{+\infty}\int_0^{2\pi} r\frac{\varepsilon^2}{2\pi}re^{-\varepsilon r}drd\theta \\ &= \frac{2}{\varepsilon} \end{aligned} . \quad (1)$$

Recall that *Information Loss* is the loss caused by representing all locations in a group with a location. It indicates the absolute value of the difference between the distance from a certain worker's true location to a certain task's obfuscated location and the distance from this worker's true location to this task's true location. We give the following example to make it be more clearer.

***Example* 1.** *As shown in Fig. 4, let μ_k denote the group center for the k-th group and $\mu_k^{'}$ denote the group center after adding noise. For a certain task t_j and the worker w_i, if we do not adopt group-based noise addition, the travel distance is $d(w_i, t_j)$. After we adopt group-based noise addition, the travel distance is $d\left(w_i, \mu_k^{'}\right)$. Thus, Information Loss, which is measured by the distance variation, can be quantified by $\left|d\left(w_i, \mu_k^{'}\right) - d(w_i, t_j)\right|$.*

Definition 3 (*Information Loss*)**.** *Formally, for a certain group C_k, Information Loss is,*

$$E_{IL}(C_k) = \sum\nolimits_{t_j \in C_k} \left|d\left(w_i, \mu_k^{'}\right) - d(w_i, t_j)\right|. \quad (2)$$

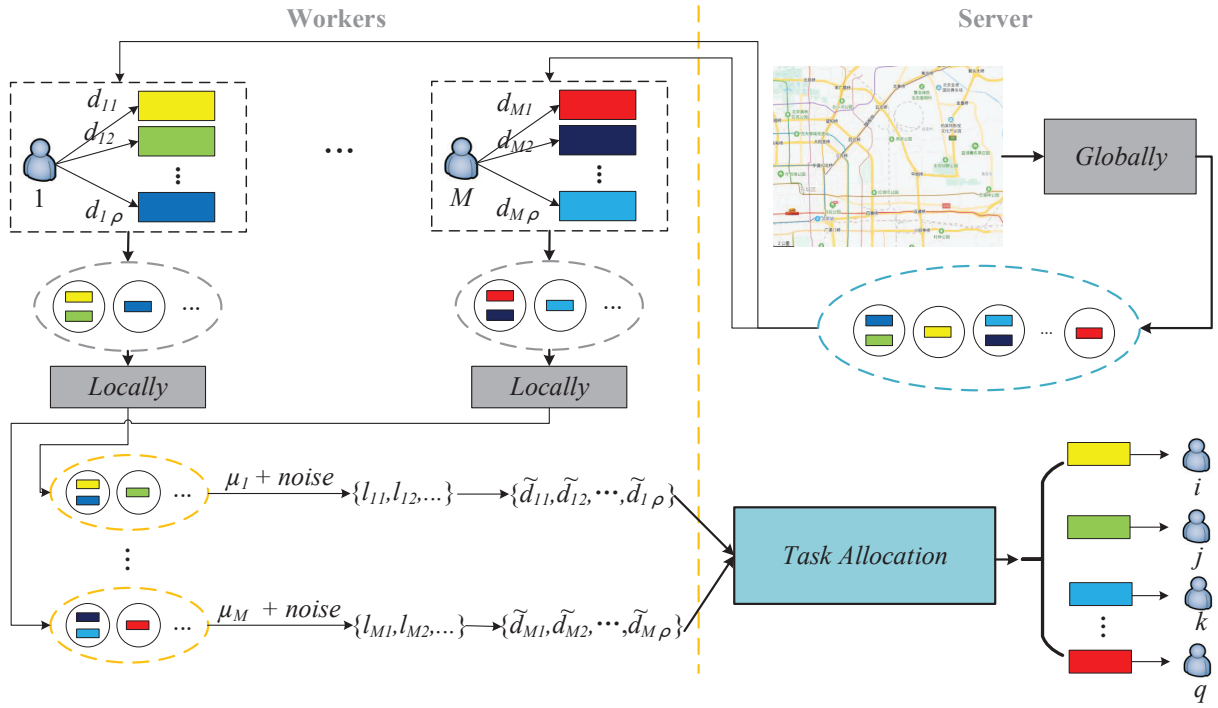


Figure 2. Overview of *CANOE*

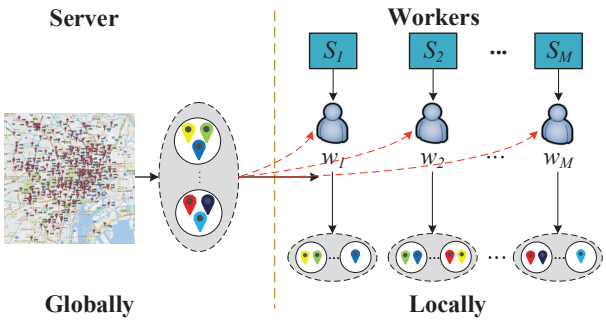


Figure 3. Overview of *OGAL*

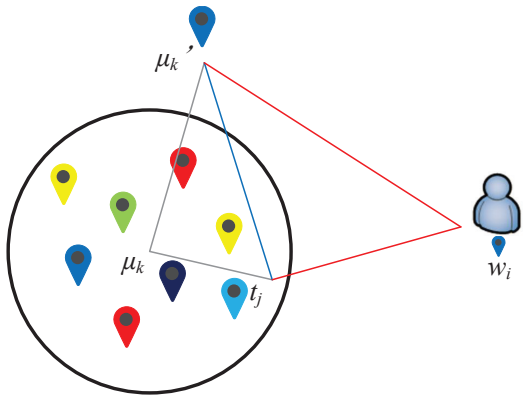


Figure 4. An example to illustrate *Information Loss*

Specifically, from Fig. 4, by triangular inequality, we have:

$$\begin{aligned} \left|d\left(w_i,\mu_k^{'}\right)-d\left(w_i,t_j\right)\right| &\le d\left(t_j,\mu_k^{'}\right) \\ d\left(\mu_k,\mu_k^{'}\right)+d\left(\mu_k,t_j\right) &\ge d\left(t_j,\mu_k^{'}\right) \end{aligned} . \tag{3}$$

Thus, we have:

$$\begin{aligned} E_{IL}\left(C_k\right) &= \textstyle\sum_{t_j\in C_k}\left|d\left(w_i,\mu_k^{'}\right)-d\left(w_i,t_j\right)\right| \\ &\le \textstyle\sum_{t_j\in C_k}\left[d\left(\mu_k,\mu_k^{'}\right)+d\left(\mu_k,t_j\right)\right] \\ &= \textstyle\sum_{q=1}^{|C_k|} d\left(\mu_k,t_q\right)+|C_k|\,d\left(\mu_k,\mu_k^{'}\right) \end{aligned} . \tag{4}$$

We consider the sum of the *Injected Noise* and *Information Loss* as the total noise, which is shown in Lemma 1.

Lemma 1. *Let $|C|$ denote the number of groups for a worker, the total noise after grouping can be quantified as follows:*

$$E\left(total\right)=\frac{2|C|^2}{\varepsilon}+\sum\nolimits_{C_k\in C}\left(\frac{2\,|C|\,|C_k|^2}{\varepsilon}\right).$$

Proof. By summing up the *Injected Noise* and *Information Loss*, we have:

$$\begin{aligned} E\left(total\right) &= \textstyle\sum_{C_k\in C}\left[E_G\left(C_k\right)+E_{IL}\left(C_k\right)\right] \\ &= \textstyle\sum_{C_k\in C}\left[\frac{2}{\frac{\varepsilon}{|C|}}+\sum_{q=1}^{|C_k|} d\left(\mu_k,t_q\right)+|C_k|\,d\left(\mu_k,\mu_k^{'}\right)\right] \\ &\le \textstyle\frac{2|C|^2}{\varepsilon}+\sum_{C_k\in C}\left[\frac{2|C_k|(|C_k|-1)}{\frac{\varepsilon}{|C|}}+|C_k|\,\frac{2}{\frac{\varepsilon}{|C|}}\right] \\ &= \textstyle\frac{2|C|^2}{\varepsilon}+\sum_{C_k\in C}\left(\frac{2|C||C_k|^2}{\varepsilon}\right) \end{aligned} .$$

□

Actually, we should make the total noise as small as possible, and give the following theorem.

Theorem 2. *The number of tasks in each group should be as near equal size as possible.*

Proof. Let Q represent the number of groups. Following Lemma 1, by the *Cauchy-Buniakowsky-Schwarz* inequality, we

have:

$$\begin{aligned}&\frac{2Q^2}{\varepsilon}+\sum_{C_k\in C}\left(\frac{2|C||C_k|^2}{\varepsilon}\right)\\&=\frac{2Q^2}{\varepsilon}+\\&\frac{2}{\varepsilon}\left[\left(|C_1|^2+|C_2|^2+\cdots+|C_Q|^2\right)\left(\overbrace{1^2+1^2+\cdots+1^2}^{Q}\right)\right]\\&\geq\frac{2Q^2}{\varepsilon}+\frac{2}{\varepsilon}\left(|C_1|+|C_2|+\cdots+|C_Q|\right)^2\end{aligned}.$$

When $|C_1|=|C_2|=\cdots=|C_Q|$, the above inequality achieves the minimum value. Thus, the size of each group should be as evenly as possible. □

5.1.2 Global Grouping Determination

According to Theorem 2, we should assure the uniformity for the size of each group. To this end, we design a balance-aware grouping method *BAG* by formalizing a constrained optimization problem. Specifically, suppose there are $|C|$ groups and the number of tasks in the k-th group is $|C_k|$. The number of tasks in each group is $\frac{|C|}{N}$ in the case of balance. Thus, we should minimize the following objective function

$$\begin{aligned}&\sum_{k=1}^{|C|}\left(|C_k|-\tfrac{|C|}{N}\right)^2\\&=\sum_{k=1}^{|C|}|C_k|^2-\tfrac{2|C|}{N}\sum_{k=1}^{|C|}|C_k|+\tfrac{|C|^3}{N^2}\\&=\sum_{k=1}^{|C|}|C_k|^2+\tfrac{|C|^3}{N^2}-2|C|\end{aligned}. \tag{5}$$

Since $|C|$ and N are constants, we should minimize $\sum_{k=1}^{|C|}|C_k|^2$. Thus, we have the following optimization problem

$$\arg\min_{u,\mu}\sum_{i=1}^{N}\sum_{k=1}^{|C|}u_{ik}d\left(t_i,\mu_k\right)+\sum_{k=1}^{|C|}\left(\sum_{i=1}^{N}a_{ik}\right)^2 \quad s.t.\begin{cases}u_{ik}\geq 0\\ \sum_{k=1}^{C}u_{ik}=1, i=1,2\cdots N\\ a_{ik}=\begin{cases}1 & k=\arg\max\left\{u_{ij}\right\}_{j=1}^{|C|}\\ 0 & other\end{cases}\end{cases}. \tag{6}$$

In Eq. 6, the first part of the objective function aims at minimizing the intra-group distances, and the second part corresponds to balance constraint, where μ_{ik} is the membership function that indicates how much the i-th sample belongs to the k-th group, and $\sum_{k=1}^{|C|}\left(\sum_{i=1}^{N}a_{ik}\right)^2=\sum_{k=1}^{|C|}|C_k|^2$. The first two constraints, denoted by *Non-negative Constraint* and *Normalizing Constraint* respectively, represent that the memberships are not negative and the sum of memberships is 1. These two jointly indicate that each task belongs to one group in C. The third constraint, denoted by *Boolean constraint*, represents that each task belongs to only one group.

We note that, this formulated problem is a mixed-integer non-linear program (*MINLP*) problem with a non-convex constraint, which is denoted by *Boolean constraint* with respect to the membership u and grouping center μ. It is NP-hard [49]. While existing non-linear optimization techniques can only deal with convex objectives effectively. Thus, a specialized solution is required to solve this non-convex *MINLP* [50].

Based on *Benders Decomposition* [51], we devise an alternate optimization method with convergence guarantee. Our overall strategy is to decompose Eq. 6 into two modules. Each module corresponds to optimize one variable while fixing the other. Thus, we can iteratively solve the two modules until convergence.

At the beginning, we need to enforce the *Boolean Constraint*. Motivated by the alternating direction multiplier method (*ADMM*) [19], we introduce the auxiliary matrix X, where an element in X is a_{ik} in Eq. 6. Thus, the objective function in Eq. 6 can be converted to

$$\arg\min_{u,\mu,X}\sum_{i=1}^{N}\sum_{k=1}^{|C|}u_{ik}d\left(t_i,\mu_k\right)+X^TX. \tag{7}$$

Thus, we alternately optimize u,μ and X.

u,μ-subproblem

When fixing X, the objective function of u,μ-subproblem is shown as follows:

$$\arg\min_{u,\mu}\sum_{i=1}^{N}\sum_{k=1}^{|C|}u_{ik}d\left(t_i,\mu_k\right)+X^TX \quad s.t.\begin{cases}u_{ik}\geq 0\\ \sum_{k=1}^{C}u_{ik}=1, i=1,2\cdots N\end{cases}. \tag{8}$$

We deal with the *Non-negative Constraint* and the *Normalizing Constraint* jointly and transform them into a *Exponential Constraint*:

$$\sum_{k=1}^{|C|}e^{-u_{ik}}=1, i=1,2\cdots N. \tag{9}$$

Thus, we have

$$\arg\min_{u,\mu}\sum_{i=1}^{N}\sum_{k=1}^{|C|}u_{ik}d\left(t_i,\mu_k\right)+X^TX \quad s.t.\quad\sum_{k=1}^{|C|}e^{-u_{ik}}=1, i=1,2\cdots N. \tag{10}$$

Obviously, such *Exponential Constraint* can replace the *Non-negative Constraint* and the *Normalizing Constraint* together as the output of the exponential function is non-negative. Accordingly, we give the lagrangian of Eq. 10:

$$\begin{aligned}L\left(u,\mu,\lambda\right)=&\sum_{i=1}^{N}\sum_{k=1}^{|C|}u_{ik}d\left(t_i,\mu_k\right)+X^TX\\&+\sum_{i=1}^{N}\lambda_i\left(\sum_{k=1}^{|C|}e^{-u_{ik}}-1\right)\end{aligned}.$$

Let the first-order derivative of lagrangian with respect to μ_k be 0, we can get:

$$\mu_k=\frac{\sum_{i=1}^{N}u_{ik}\cdot t_k}{\sum_{i=1}^{N}u_{ik}}. \tag{11}$$

Let the first-order derivative of lagrangian with respect to u_{ik} be 0, with the *Exponential constraint*, we can get:

$$u_{ik}=-\ln\frac{t_i-\mu_k}{\sum_{k=1}^{|C|}\left(t_i-\mu_k\right)}. \tag{12}$$

X-subproblem

When fixing u and μ, the objective function of X-subproblem is shown as follows:

$$\arg\min_{X}X^TX \quad s.t.\quad a_{ik}=\begin{cases}1 & k=\arg\max\left\{u_{ij}\right\}_{j=1}^{|C|}\\ 0 & other\end{cases}. \tag{13}$$

To deal with the *Boolean Constraint*, we introduce an auxiliary matrix Z, and the above objective function is further transformed into:

$$\arg\min_{X}X^TX \quad s.t.\quad X-Z=0. \tag{14}$$

Based on *ADMM*, the above equation is transformed into

Algorithm 1: Balance-aware Grouping

Input: $|C|$: the number of groups;
T: the set of tasks;
τ: the scaling factor;
Output: X: the groupings;

initialize u and μ and $U = 0$;
initialize X according to u;
while *not converge* **do**
  update μ_k using Eq. 11;
  update u_{ij} using Eq. 12;
  update Z using Eq. 16;
  update X using Eq. 17;
  set $U = U + \tau(X - Z)$;
  set $\tau = 1.1\tau$;
return X;

an unconstrained optimization problem:

$$\arg\min_{Z,U,\tau} Z^T Z + \frac{\tau}{2}\left\|X - Z + \frac{1}{\tau}U\right\|_F^2, \tag{15}$$

where $\tau > 0$ is a scaling factor and U is the lagrange multiplier.

Let the first-order derivative of lagrangian with respect to Z be 0, we can get:

$$Z = (2I + \tau)^{-1} \cdot (\tau X + U), \tag{16}$$

where every element in I is 1 and I has the same scale with X.

When Z, U and τ are fixed, the optimal solution of X is equivalent to

$$X = \min \left\|X - Z + \frac{1}{\tau}U\right\|_F^2.$$

Thus, we have:

$$X_{ik} = \begin{cases} 1 & k = \arg\max\{v_{ij}\}_{j=1}^{|C|} \\ 0 & other \end{cases}, \tag{17}$$

where $V = Z - \frac{1}{\tau}U$.

The overall procedure is shown in Alg. 1.

5.1.3 Convergence Analysis

To prove the convergence of Alg. 1, we need to prove that the above two subproblems are monotonically bounded.

Lemma 2. *The objective function in Eq. 15 is non-increasing while enlarging Z.*

Proof. We only need to prove that the *Hessian Matrix* about Z is positive definite. We have *Hessian Matrix*:

$$\begin{aligned}\Delta^2 J(Z) &= \begin{bmatrix} \frac{\partial^2 J(Z)}{\partial Z_{11}\partial Z_{11}} & \cdots & \frac{\partial^2 J(Z)}{\partial Z_{11}\partial Z_{N|C|}} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 J(Z)}{\partial Z_{N|C|}\partial Z_{11}} & \cdots & \frac{\partial^2 J(Z)}{\partial Z_{11}\partial Z_{11}} \end{bmatrix}. \\ &= \begin{bmatrix} 2+\tau & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 2+\tau \end{bmatrix}\end{aligned}$$

Obviously, $\Delta^2 J(Z) > 0$. Thus, *Hessian Matrix* is positive definite.

This concludes the proof. □

Lemma 3. *The objective function in Eq. 15 is bounded.*

Proof. Obviously,

$$\frac{\tau}{2}\left\|X - Z + \frac{1}{\tau}U\right\|_F^2 \geq 0. \tag{18}$$

As for $Z^T Z$, it is equivalent to the following optimization problem:

$$\begin{aligned}&\arg\min_x \sum_{k=1}^{|C|} x_k^2 \\ &s.t. \sum_{k=1}^{|C|} x_k = N\end{aligned}. \tag{19}$$

By the *Cauchy-Buniakowsky-Schwarz* inequality, we have:

$$\sum_{k=1}^{|C|} x_k^2 \left(1^2 + \cdots + 1^2\right) \geq \left(x_1 + \cdots + x_{|C|}\right)^2 = N^2. \tag{20}$$

Thus, we have:

$$\sum_{k=1}^{|C|} x_k^2 \geq \frac{N^2}{|C|}. \tag{21}$$

Thus, the objective function in Eq. 15 is greater than $\frac{N^2}{|C|}$.

This concludes the proof. □

Theorem 3. *By Benders Decomposition and ADMM, the problem in Eq. 6 can at least convergence to a local optimum.*

Proof. As for u, μ-subproblem, since an arbitrary norm on $\mathbb{R}$ is convex, non-negative summation and compound affine mapping are convex-preserving operations, by the lagrange multiplier approach, the problem shown in Eq. 10 can at least convergence to a local optimum.

As for X-subproblem, according to Lemma 2 and Lemma 3, the problem shown in Eq. 15 can also convergence to a local optimum.

□

5.2 Adaptive Local Adjustment

Given the global groupings, we now design an adaptive local adjustment method *ALA*. The underlying idea is to achieve fine-grained partitions over dense groups and coarse-grained partitions over sparse groups to minimize the total noise. In this way, each worker can adaptively determine the optimal grouping, including the number of tasks in each group and the number of groups. Specifically, we first derive another constraint. Then, based on the new derived constraint, as well as the constraint in Theorem 2, we formalize two optimization problems to enforce these two constraints jointly.

Actually, we should make the total noise of *CANOE* be less than that of adding noise to each task's location in a worker's preference set. To achieve this goal, suppose for a certain group, the number of tasks is $|C_k|$. The privacy budget for each task is $\varepsilon' = \frac{\varepsilon}{|C_k|}$ if we add noise independently. In this case, the *Information Loss* is the change of travel distance only caused by adding noise. Thus, the *Information Loss* for the k-th group is $|C_k|\frac{2}{\varepsilon'}$. To make our group-based noise addition be better than the above method, we let the $E_{IL}(C_k) \leq |C_k|\frac{2}{\varepsilon'}$, and get the following theorem.

Theorem 4. *Let μ and $|C_k|$ denote the group center and the number of tasks for the k-th group respectively. Additionally, let t_q denote a task in the group, and the sum of the inner distance of this group should satisfy the following constraint,*

$$\sum_{q=1}^{|C_k|} d(\mu, t_q) \leq \frac{2|C_k|(|C_k| - 1)}{\varepsilon}.$$

Proof. Let $E_{IL}(C_k) \leq |C_k|\frac{2}{\varepsilon'}$, we have:

$$\begin{aligned}\sum_{q=1}^{|C_k|} d(\mu, t_q) &\leq \frac{2|C_k|^2}{\varepsilon} - |C_k|\, d\left(\mu, \mu_k'\right) \\ &= \frac{2|C_k|^2}{\varepsilon} - |C_k|\frac{2}{\varepsilon} \\ &= \frac{2|C_k|(|C_k|-1)}{\varepsilon}\end{aligned}.$$

□

For convenience, we denote the constraint in Theorem 2 by *Balance Constraint*, and denote the constraint in Theorem 4 by *Intra-group Constraint*. We need to meet them jointly as much as possible. Since, obviously, the fewer elements in a group, the *Intra-group Constraint* can be more likely satisfied, and it is conducive to the realization of *Balance Constraint*. Therefore, our overall strategy is presented as follows. First, we deal with *Intra-group Constraint* and design a dynamic programming based method by minimizing the maximum sum within each group of a worker. Then, we design a migration-based adjustment method to achieve *Balance Constraint* through as few times as possible.

5.2.1 Achieving Intra-group Constraint

Obviously, if the maximal sum of distance within each group satisfies the *Intra-group Constraint*, we can easily enforce the *Intra-group Constraint* for each group of a worker. To this end, we first formulate the following problem by determining the minimized maximal *Intra-group* distance.

Problem 1. *Given a non-negative structure set G. Every element G_i in G includes a group C_k and two attributes $G_i.d_1$ and $G_i.d_2$, where the two attributes indicate $\sum_{q=1}^{|C_k|} d(\mu, t_q)$ and $\frac{2|C_k|(|C_k|-1)}{\varepsilon}$ respectively. Each group in G has been sorted according to the relative distance from the true location of each worker. We aim to merge $|G|$ groups into $|C|$ non-empty continuous groups, and minimize maximal d_1.*

We give the following example to illustrate the above problem.

***Example* 2.** *Suppose we have 5 groups, which form an ordered array $[8, 2, 4, 9, 8]$. Thus, we have $|G|$=5. Suppose $|C|$=2. The merged groups are $[8, 2, 4]$ and $[9, 8]$ as the maximum sum is 17, which is the smallest in all cases. Note that, $[8, 2, 9]$ is not a valid group as the elements 2 and 9 in the original ordered array are not adjacent.*

To solve the above problem, we design a method to **M**inimize the **M**aximum **S**um based on dynamic programming, called *MMS*. In particular, let f_k^r denote the maximum value when merging the first k groups into r groups, the state transfer equation is shown as follows:

$$f_k^r = \max\left(f_j^{r-1}, \sum_{q=j+1}^{k} G_q.d_1\right). \tag{22}$$

It indicates that the maximum value of the *Intra-group* distance from the first k groups ($k \leq |G|$) when merged into r ($r \leq |C|$) groups, is the larger of the value, when merging j groups ($j \leq k$) into $r-1$ groups and merging the $j+1$-th to k-th groups into only one group. Each worker traverses all k and r, and returns

$$\theta = f_{|G|}^{|C|}. \tag{23}$$

Alg. 2 shows the main steps of *MMS*. In *MMS*, with the initial groupings set C, each worker first splits each group into two groups, where a random location is taken out. Thus, $|G| = 2|C|$ (Line 1). Then, he sorts G and calculates corresponding d_1 and d_2. Specifically, for sorting, the group center that is nearest to the worker's true location is ranked first, and the center of another one group who is nearest to the above group center is ranked second. It follows the same manner for the remaining group centers. In this way, the sorting can be completed (Line 2). Next, each worker initializes every element of f with positive infinity except for $f_0^0 = 0$ (Line 3). After that, he solves the problem in Eq. 22 and gets θ (Lines 4-8). Finally,

Algorithm 2: MMS

Input: C: the initial grouping for a worker;
Output: C: the adjusted grouping for a worker;

1 split C;
2 initialize and sort G;
3 initialize f;
4 **for** $i = 1$ to $|G|$ **do**
5 **for** $j = 1$ to $|C|$ **do**
6 **for** $k = 1$ to $k < i$ **do**
7 set
$$f_i^j = \min\left\{f_i^j, \max\left(f_k^{j-1}, \sum_{q=k+1}^{i} G_q.d_1\right)\right\};$$
8 set $\theta = f_{|G|}^{|C|}$;
9 merge G to C according to θ and *Intra-group Constraint*;
10 return C;

he merges a group until $d_1 > d_2$ or $d_1 > \theta$ (Line 9). It follows the same manner for the remaining workers.

5.2.2 Achieving Balance Constraint

After tackling *Intra-group Constraint* by *MMS*, we now enforce the *Balance Constraint* through determining the optimal number of tasks in each group. According to Lemma 1, when groupings are relatively balanced, we need to keep the number of groups as small as possible. Moreover, after *MMS*, there may be too many locations in some groups and too less locations in other groups. Thus, we migrate tasks among groups to adjust them, and propose a **M**igration-based **G**rouping **A**djustment method *MGA*.

To reduce the cost, that is the number of migrations, when conducting local adjustment, we first formalize the following problem.

Problem 2. *Given an array with non-negative integers $[x_1, x_2 \cdots x_{|C|}]$ ($x_i \in \mathbb{Z}^+$ and $\sum_{i=1}^{|C|} x_i = \rho$), where ρ is the number of elements in the preference set. We aim to find the minimum number of migration required to make all elements of the array be equal as much as possible. That is,*

$$\arg\min_x \sum_{i=1}^{|C|} |x - x_i|.$$

We then give the following theorem to tackle the above problem.

Theorem 5. *For Prob. 2, the optimal solution is its median.*

Proof. Suppose $x_1 \leq x_2 \ldots \leq x_{|C|}$. Let $h(x)$ and x^* denote the objective function and the corresponding optimal solution respectively.

If $h(x) = |x - x_1| + |x - x_{|C|}|, x \in [x_1, x_{|C|}]$, $x^* = x_{|C|}$.

If $h(x) = |x - x_2| + |x - x_{|C|-1}|, x \in [x_1, x_{|C|-1}]$, $x^* = x_{|C|-1}$.

If $h(x) = |x - x_k| + |x - x_{k+1}|, x \in [x_k, x_{k+1}]$, $x^* = x_{k+1}$.

According to the *Mathematical Methods of Induction*, since $[x_k, x_{k+1}] \subseteq \ldots \subseteq [x_2, x_{|C|-1}] \subseteq [x_1, x_{|C|}]$, the optimal solution is its median x_{k+1}. Specifically, we have

$$x^* = \begin{cases} \{x_k, x_{k+1}\} & |C| = 2k \\ x_{k+1} & |C| = 2k+1 \end{cases}. \tag{24}$$

This concludes the proof. □

Alg. 3 shows the main steps of *MGA*. In *MGA*, by feeding the grouping obtained by *MMS*, each worker first calculates

Algorithm 3: MGA

Input: C: the grouping obtained by *MMS*;
Output: C: the adjusted grouping for a worker;
1 calculate the number of elements in a group ;
2 calculate the median $|C|^*$;
3 **for** each group C_k in C **do**
4 **if** $|C_k| < |C|^*$ **then**
5 merge the locations from the near group or groups;
6 return C;

the number of elements in a group (Line 1). Then, he calculates the median $|C|^*$ according to Eq. 24 (Line 2). Next, for each group C_k in C, a worker finds the nearest one or more groups in which the number of elements is greater than $|C|^*$, and migrates elements from it or them. The migration stops once satisfying the *Intra-group Constraint* or the number of elements in the migrated groups is larger than $|C|^*$ (Lines 3-4). Finally, he obtains C (Line 5).

6 Utility-aware Obfuscated Distance Collection

After obtaining the optimized grouping for each worker, he adds noise for each group to collect and uploads the required distances for subsequent task allocation. Specifically, we first obtain a representative location for each group, and then add noise for this location. A straightforward approach is to regard the average location within each group as the representative location and calculate the distances from a worker's true location to the obfuscated representative location while adopting *PL*. However, we note that each generated representative location for a certain group should be close to tasks' locations in the same group and be far away from tasks' locations in other groups. In addition, the straight-forward adoption of *PL* may introduce excessive noise due to the randomness and boundlessness of the laplacian distribution.

To this end, we propose a utility-aware obfuscated distance collection method *UODC* with solid privacy and utility guarantees by keeping each generated representative location within the range formed by the tasks' locations in the same group. It mainly consists of two phases. In the first phase, we formalize an optimization problem by minimizing intra-group distances and maximizing inter-group distances. In the second phase, we let each worker generate a noisy location for each representative location, and finally uploads the calculated noisy distance from his true location to each noisy representative location to the server for subsequent task allocation. We prove that *UODC* satisfies Geo-I and show its utility superiority with solid utility analysis.

In the following, we first present the details of the two phases. Then, we give the privacy and utility guarantees of *UODC*.

6.1 Design of *UODC*

In the first phase, to generate a representative location μ_k for the group C_k, we formalize the following optimization problem

$$\arg\min_{\mu_k} \sum_{i=1,t_i\in C_k}^{|C_k|} (\mu_k - t_i)^2 - \sum_{h=1,h\neq k}^{|C|} \left(\mu_k - \frac{\sum_{j=1,t_j\in C_h,}^{|C_h|} t_j}{|C_h|} \right)^2 . \tag{25}$$

In Eq. 25, the first part of the objective function indicates that we make μ_k be close to t_i, which is also in the group C_k. The second part indicates that μ_k should stay as far away from the tasks' locations in other groups as possible, where $|C|$ is the number of groups for a worker. To minimize the objective function in Eq. 25, we should make the partial derivative be equal to 0, we have

$$\mu_k = \frac{\sum_{h=1,h\neq k}^{|C|} \sum_{j=1,t_j\in C_h,}^{|C_h|} t_j - \sum_{i=1,t_i\in C_k}^{|C_k|} t_i}{|C| - |C_k| - 1}. \tag{26}$$

In the second phase, we note that if a location is far away from the true location, the probability that it is regarded as the obfuscated location should be decreased. Thus, we let the probability of sampling the obfuscated location $\mu_k^{'}$ within C_k be negatively correlated with the distance between $\mu_k^{'}$ and μ_k. Therefore, we generate $\mu_k^{'}$ by Eq. 27.

$$P\left(\mu_k^{'} | \mu_k\right) = \frac{\varepsilon \cdot e^{-\frac{\varepsilon}{2} d\left(\mu_k, \mu_k^{'}\right)}}{\sum_{z\in C_k} \varepsilon \cdot e^{-\frac{\varepsilon}{2} d(\mu_k, z)}}. \tag{27}$$

Algorithm 4: UODC

Input: C_{li}: the grouping for the i-th worker;
l_i: the location for the i-th worker;
M: the number of workers;
ε: the privacy budget;
Output: D: the distance matrix;
1 **while** $i \leq M$ **do**
2 **for** each group C_k in C_{li} **do**
3 computes $\varepsilon' = \frac{\varepsilon}{|C_{li}|}$;
 `// determine the representative location for each group`
4 generate μ_k by Eq. 26;
 `// generate the noisy representative location`
5 sample $\mu_k^{'}$ from C_k by Eq. 27 using ε';
6 compute the distance between l_i and $\mu_k^{'}$;
7 record the distance in D;
8 return D;

6.2 Analysis of *UODC*

6.2.1 Privacy Analysis

Theorem 6. *For any worker with privacy budget ε, UODC satisfies ε-Geo-I.*

Proof. By definition, to prove *UODC* satisfies ε-Geo-I, we need to prove

$$\frac{P(z|x)}{P(z|y)} \leq e^{\varepsilon d(x,y)} .$$

Since $P(z|x) = \frac{\varepsilon \cdot e^{-\frac{\varepsilon}{2} d(x,z)}}{\sum_{v\in V} \varepsilon \cdot e^{-\frac{\varepsilon}{2} d(x,v)}}$, by the triangular inequality, we have

$$\begin{aligned}
\frac{P(z|x)}{P(z|y)} &= \frac{\varepsilon \cdot e^{-\frac{\varepsilon}{2} d(x,z)}}{\varepsilon \cdot e^{-\frac{\varepsilon}{2} d(y,z)}} \cdot \frac{\sum_{v\in V} \varepsilon \cdot e^{-\frac{\varepsilon}{2} d(y,v)}}{\sum_{v\in V} \varepsilon \cdot e^{-\frac{\varepsilon}{2} d(x,v)}} \\
&\leq e^{\frac{\varepsilon}{2}(d(y,z) - d(x,z))} \cdot \frac{\sum_{v\in V} \varepsilon \cdot e^{\frac{\varepsilon}{2} d(y,x) - \frac{\varepsilon}{2} d(x,v)}}{\sum_{v\in V} \varepsilon \cdot e^{-\frac{\varepsilon}{2} d(x,v)}} \\
&\leq e^{\frac{\varepsilon}{2} d(x,y)} \cdot e^{\frac{\varepsilon}{2} d(x,y)} \\
&\leq e^{\varepsilon d(x,y)}
\end{aligned} .$$

This concludes the proof. □

6.2.2 Utility Analysis

Theorem 7. *The average error and maximum error of UODC are both less than PL.*

Proof. We adopt E to denote the average error of *UODC*, which can be quantified by the weighted distance from each possible obfuscated location to the true location. The weight can be defined by the probability of reporting a possible obfuscated location according to the true location. Additionally, suppose there are n tasks in a group. Thus, we have:

$$\begin{aligned} E\left[d\left(\mu,t\right)\right] &= \sum_{i=1}^{n} p\left(t_i\left|\mu\right.\right)\cdot d\left(\mu,t_i\right) \\ &= \sum_{i=1}^{n} \frac{\varepsilon e^{-\frac{\varepsilon}{2}\cdot d\left(\mu,t_j\right)}}{\sum\limits_{j=1}^{n}\varepsilon e^{-\frac{\varepsilon}{2}\cdot d\left(\mu,t_j\right)}}\cdot d\left(\mu,t_i\right) \end{aligned}. \tag{28}$$

For the computational purpose, we use the constant d_0 and a secondary variable o_i to represent $d\left(\mu,t_i\right)$, where d_0 denotes the maximum distance between μ and t_i, and o_i denotes the contribution of t_i for μ. Moreover, since the larger o_i, the smaller $d\left(\mu,t_i\right)$. Thus, we set

$$d\left(\mu,t_i\right) = \left(1-o_i\right)\cdot d_0. \tag{29}$$

Hence, we have:

$$\begin{aligned} E\left[d\left(\mu,t\right)\right] &= d_0\sum_{i=1}^{n}\frac{\left(1-o_i\right)e^{-\frac{\varepsilon}{2}d_0\left(1-o_i\right)}}{\sum\limits_{j=1}^{n}e^{-\frac{\varepsilon}{2}d_0\left(1-o_j\right)}} \\ &= d_0\frac{\sum\limits_{i=1}^{n}\left(1-o_i\right)e^{\frac{\varepsilon}{2}d_0o_i}}{\sum\limits_{j=1}^{n}e^{\frac{\varepsilon}{2}d_0o_j}} \\ &= d_0\left(1-\frac{o_1e^{\frac{\varepsilon}{2}d_0o_1}+o_2e^{\frac{\varepsilon}{2}d_0o_2}+\cdots+o_ne^{\frac{\varepsilon}{2}d_0o_n}}{e^{\frac{\varepsilon}{2}d_0o_1}+e^{\frac{\varepsilon}{2}d_0o_2}+\cdots+e^{\frac{\varepsilon}{2}d_0o_n}}\right) \end{aligned}. \tag{30}$$

According to the *Taylor Expansion* [52], we have $e^x \geq x+1$. Thus, we have

$$\begin{aligned} \sum_{i=1}^{n} o_ie^{\frac{\varepsilon}{2}d_0o_i} &\geq \sum_{i=1}^{n}\left(\tfrac{\varepsilon}{2}d_0o_i+1\right)o_i \\ &= \tfrac{\varepsilon}{2}d_0\sum_{i=1}^{n}o_i^2+\sum_{i=1}^{n}o_i \end{aligned}. \tag{31}$$

According to the *Law of Large Numbers* [53], we have $\sum\limits_{i=1}^{n}x_i = n\cdot E\left(x\right)$ and $E\left(x^2\right) = E^2\left(x\right)+D\left(x\right)$, where $E\left(x\right)$ and $D\left(x\right)$ are the mean and variance respectively. Moreover, since there is no any prior information, according to the *Maximum Entropy Theory*, we assume o is sampled from a uniform distribution with range $(0,1)$. Thus, the mean and variance of o are $\frac{1}{2}$ and $\frac{1}{12}$ respectively. Therefore, we have:

$$\sum_{i=1}^{n} o_ie^{\frac{\varepsilon}{2}d_0o_i} \geq n\left(\frac{\varepsilon}{2}\cdot\frac{1}{3}d_0+\frac{1}{2}\right). \tag{32}$$

Moreover, to obtain $\sum\limits_{i=1}^{n}e^{\frac{\varepsilon}{2}d_0o_i}$, we formulate the following problem.

Problem 3. *Given $o\sim U\left(0,1\right)$, what is the probability density function of $e^{\frac{\varepsilon}{2}d_0o}$?*

To solve Prob. 3, let an auxiliary variable $h = e^{\frac{\varepsilon}{2}d_0o}$. Thus, we have

$$\begin{aligned} P\left(H\leq h\right) &= P\left(e^{\frac{\varepsilon}{2}d_0O}\leq h\right) \\ &= P\left(0\leq O\leq\tfrac{2\ln h}{\varepsilon d_0}\right) \\ &= \int_0^{\frac{2\ln h}{\varepsilon d_0}}1\cdot dO \\ &= \tfrac{2\ln h}{\varepsilon d_0} \end{aligned}.$$

Since $P'\left(H\leq h\right) = \frac{2}{\varepsilon d_0 h}$, we have

$$pdf\left(h\right) = \begin{cases}\frac{2}{\varepsilon d_0 h} & 1\leq h\leq e^{\frac{\varepsilon}{2}d_0}\\ 0 & otherwise\end{cases}.$$

Thus, we have

$$E\left(h\right) = \frac{2\left(e^{\frac{\varepsilon}{2}d_o}-1\right)}{\varepsilon d_0}.$$

Therefore. we have

$$\begin{aligned} E\left[d\left(\mu,t\right)\right] &\leq d_0\left(1-\frac{n\left(\frac{\varepsilon}{6}d_0+\frac{1}{2}\right)}{n\cdot E(h)}\right) \\ &= d_0\left(1-\frac{\varepsilon d_o n\left(\frac{\varepsilon}{6}d_0+\frac{1}{2}\right)}{2\left(e^{\frac{\varepsilon}{2}d_o}-1\right)}\right) \\ &\approx d_0\left(\tfrac{1}{2}-\tfrac{\varepsilon}{6}d_0\right) \end{aligned}.$$

Since the expected error of *PL* is $E_{PL} = \frac{2}{\varepsilon}$, in what follows, we need to compare $E\left[d\left(\mu,t\right)\right]$ with E_{PL}. We construct $g\left(\varepsilon,d_0\right) = E\left[d\left(\mu,t\right)\right]-E_{PL}$. If $g\left(\varepsilon,d_0\right)<0$, we can claim that the average error of *UODC* is less than *PL*. Specifically, we have

$$\begin{aligned} g\left(\varepsilon,d_0\right) &= d_0\left(\tfrac{1}{2}-\tfrac{\varepsilon}{6}d_0\right)-\tfrac{2}{\varepsilon} \\ &= d_0\left(3-\varepsilon d_0\right)-\tfrac{12}{\varepsilon} \\ &= 3d_0-\tfrac{12}{\varepsilon}-\varepsilon d_0^2 \end{aligned}. \tag{33}$$

Since $\frac{12}{\varepsilon}+\varepsilon d_0^2 > 2\sqrt{12\cdot d_0^2} = 4\sqrt{3}d_0$, we can claim $g\left(\varepsilon,d_0\right)<0$ for any ε and d_0.

As for the maximum error, the maximum error of *UODC* is d_0, which is limited while the maximum error of *PL* is positive infinity due to the boundlessness of the laplacian distribution. Thus, the maximum error of *UODC* is also less than *PL*.

□

7 Analysis of *CANOE*

In this Section, we theoretically analyze the privacy, utility and complexity of *CANOE*.

7.1 Privacy Guarantee

At beginning, we give Lemma 4 to quantify the distance between any two locations after adding noise based on our group-based noise addition.

Lemma 4. *For any two tasks' locations with the distance less than r ($d\left(l_1,l_2\right)\leq r$), the distance is still less than r after they are grouped. For notation convenience, the grouping method is denoted by $\mathcal{G}$. That is, it holds $d\left(\mathcal{G}\left(l_1\right),\mathcal{G}\left(l_2\right)\right)\leq r$.*

Proof. Let x and y represent the longitude and latitude of a task respectively. We have

$$\mathcal{G}\left(l_1\right) = \left(\frac{\sum\limits_{i=1}^{m}x_i}{m}+r\cdot\cos\theta,\frac{\sum\limits_{i=1}^{m}y_i}{m}+r\cdot\sin\theta\right)$$

and

$$\mathcal{G}\left(l_2\right) = \left(\frac{\sum\limits_{j=1}^{n}x_j}{n}+r\cdot\cos\theta,\frac{\sum\limits_{j=1}^{n}y_j}{n}+r\cdot\sin\theta\right)$$

where m and n represent how many locations the two groups are formed from.

Obviously, we have

$$\begin{aligned} &d\left(\mathcal{G}\left(l_1\right),\mathcal{G}\left(l_2\right)\right) \\ &= \sqrt{\left(\frac{\sum\limits_{i=1}^{m}x_i}{m}-\frac{\sum\limits_{j=1}^{n}x_j}{n}\right)^2+\left(\frac{\sum\limits_{i=1}^{m}y_i}{m}-\frac{\sum\limits_{j=1}^{n}y_j}{n}\right)^2} \\ &= \sqrt{\frac{1}{(mn)^2}\left[\left(n\sum_{i=1}^{m}x_i-m\sum_{j=1}^{n}x_j\right)^2+\left(n\sum_{i=1}^{m}y_i-m\sum_{j=1}^{n}y_j\right)^2\right]} \end{aligned}.$$

Therefore, we have

$$d\left(\mathcal{G}\left(l_1\right),\mathcal{G}\left(l_2\right)\right) = \sqrt{\frac{1}{(mn)^2}\left\{\begin{array}{l}\left[\overbrace{\left(x_{i1}-x_{j1}\right)}^{s_1}+\cdots+\overbrace{\left(x_{i(mn)}-x_{j(mn)}\right)}^{s_{mn}}\right]^2 \\ +\left[\overbrace{\left(y_{i1}-y_{j1}\right)}^{t_1}+\cdots+\overbrace{\left(y_{i(mn)}-y_{j(mn)}\right)}^{t_{mn}}\right]^2\end{array}\right\}}.$$

Since $\left(\sum_{i=1}^{N} a_i\right)^2 \leq N\sum_{i=1}^{N} a_i^2$, we have

$$\begin{aligned} & d\left(\mathcal{G}\left(l_1\right),\mathcal{G}\left(l_2\right)\right) \\ \leq & \sqrt{\frac{1}{(mn)^2}\left\{mn\left[\left(s_1^2+t_1^2\right)+\cdots+\left(s_{mn}^2+t_{mn}^2\right)\right]\right\}} \\ \leq & \sqrt{\frac{1}{(mn)^2}mn\left(mn\cdot r^2\right)} \\ = & r \end{aligned}.$$

This concludes the proof. □

Based on Lemma 4, we give a comprehensive analysis of *CANOE* on privacy guarantee in Theorem 8.

Theorem 8. *For any worker with privacy budget ε, CANOE satisfies ε-Geo-I.*

Proof. By definition, to prove our *CANOE* satisfies ε-Geo-I, we need to prove

$$\frac{P\left(\mathcal{G}\left(l_1\right)=l^*\left|l_1\right.\right)}{P\left(\mathcal{G}\left(l_2\right)=l^*\left|l_2\right.\right)} \leq e^{\varepsilon d(l_1,l_2)}.$$

Similar to Theorem 6, we have $P\left(\mathcal{G}\left(l_1\right)=l^*\left|l\right.\right) = \frac{\frac{\varepsilon}{|C|}e^{-\frac{\varepsilon}{2|C|}d\left(\mathcal{G}(l),l^*\right)}}{\sum_{z\in V}\frac{\varepsilon}{|C|}e^{-\frac{\varepsilon}{2|C|}d(\mathcal{G}(l),z)}}$. Thus, we have

$$\frac{P\left(\mathcal{G}\left(l_1\right)=l^*\left|l_1\right.\right)}{P\left(\mathcal{G}\left(l_2\right)=l^*\left|l_2\right.\right)} \leq e^{\frac{\varepsilon}{|C|}d(\mathcal{G}(l_1),\mathcal{G}(l_2))}.$$

According to Lemma 4, for each task, we have

$$\frac{P\left(\mathcal{G}\left(l_1\right)=l^*\left|l_1\right.\right)}{P\left(\mathcal{G}\left(l_2\right)=l^*\left|l_2\right.\right)} \leq e^{\frac{\varepsilon}{|C|}d(l_1,l_2)}.$$

According to sequential composition property in Theorem 1, for each worker, we have

$$\frac{P\left(\mathcal{G}\left(l_1\right)=l^*\left|l_1\right.\right)}{P\left(\mathcal{G}\left(l_2\right)=l^*\left|l_2\right.\right)} \leq e^{\varepsilon d(l_1,l_2)}.$$

This concludes the proof. □

7.2 Utility Guarantee

In what follows, we present the utility analysis to explain why *CANOE* performs better in Theorem 9.

Theorem 9. *If a worker is eventually assigned to a task, CANOE makes he be more likely to be assigned to the optimal task, which can make the average travel distance be smaller, than directly adding noise to his true location.*

Proof. Recall that we assume that the size of the preference set for each worker is the same, denoted by ρ, and each worker uses the same privacy budget ε. We introduce auxiliary variables p_1 and p_2 to denote the probabilities of being assigned to the optimal task when directly adding noise to workers' locations and adopting the proposed *CANOE* respectively.

Clearly, we have $p_1 = \frac{1}{\rho}$. For *CANOE*, we discuss p_2 in two cases.

Case 1: If there is only one group, we have $p_2 = \frac{1}{\rho}$. Thus, we have $p_2 = p_1$.

Case 2: If there are more than two groups, we first introduce auxiliary variable $a_1, a_2 ... a_i ... a_{|C|}$, where $|C|$ is the number of groups and a_i is the number of elements in the i-th group.

If the tasks in the preference set are evenly divided into $|C|$ groups, we have $p_2 = \frac{1}{|C|}\cdot\frac{1}{\rho/|C|}$. Thus, $p_2 = p_1$.

If the tasks in the preference set are not evenly divided into $|C|$ groups, we have

$$p_2 = \frac{1}{|C|}\cdot\left(\frac{1}{|C|}\cdot\frac{1}{a_1}+\frac{1}{|C|}\cdot\frac{1}{a_2}+...\frac{1}{|C|}\cdot\frac{1}{a_i}...+\frac{1}{|C|}\cdot\frac{1}{a_{|C|}}\right), \tag{34}$$

where $a_1+a_2+a_i+a_{|C|}=\rho$ and $1\leq a_i<\rho$. Thus, we have

$$\begin{aligned} |C|^2\cdot p_2 = & \frac{1}{a_1}+\frac{1}{a_2}+...\frac{1}{a_i}...+\frac{1}{a_{|C|}} \\ = & \frac{a_1+a_2+a_i+a_{|C|}}{a_1\cdot\rho}+\frac{a_1+a_2+a_i+a_{|C|}}{a_2\cdot\rho} \\ & +...\frac{a_1+a_2+a_i+a_{|C|}}{a_i\cdot\rho}...+\frac{a_1+a_2+a_i+a_{|C|}}{a_{|C|}\cdot\rho} \\ = & \frac{|C|}{\rho}+\frac{1}{\rho}\cdot\left[\left(\frac{a_2}{a_1}+\frac{a_1}{a_2}\right)+\left(\frac{a_3}{a_1}+\frac{a_1}{a_3}\right)+...\right] \end{aligned}. \tag{35}$$

Since $x+y\geq 2\sqrt{xy}$ for any $x,y\geq 0$, we have $\frac{a_j}{a_i}+\frac{a_i}{a_j}\geq 2$. Thus, we have

$$\begin{aligned} |C|^2\cdot p_2 & \geq \frac{|C|}{\rho}+\frac{2}{\rho}\cdot\frac{|C|\cdot(|C|-1)}{2} \\ & \geq \frac{|C|^2}{\rho} \end{aligned}. \tag{36}$$

Therefore, we have $p_2 \geq \frac{1}{\rho}$. Thus, $p_2 \geq p_1$.

In summary, we have $p_2 \geq p_1$.

□

7.3 Time Complexity

To show the efficiency of *CANOE*, we analyze the increased computational cost compared with the non-private task allocation approach on both the worker side and the server side. Generally, the time for balancing privacy and utility trade-off is increased while the time for striking utility and efficiency trade-off is reduced. The reasons lie in that, we have increased the extra time for conducting grouping and adjusting each grouping, but have reduced the time for calculating the needed distances due to the adoption of group-based noise addition.

Specifically, on the server side, the server participates in **Phase 1** and **Phase 4**. Since the approach in [2] need to be performed both for non-private task allocation and *CANOE* in **Phase 4**, the operations in **Phase 4** are not included. We only need to calculate the time consumption of invoking Alg. 1 in **Phase 1**. Specifically, in Alg. 1, the server first initializes u, μ, U and X (Lines 1-2), leading to $O\left(N\cdot|C|\right)$, $O\left(N\cdot|C|\right)$, $O\left(1\right)$ and $O\left(N\cdot|C|\right)$, as there are $|C|$ group centers and each initialization involves N tasks. Moreover, initialization with 0 usually can be completed in $O\left(1\right)$. Then, similarly, in each iteration, each of updates for μ_k, u_{ij}, Z, X and U also takes $O\left(N\cdot|C|\right)$. Additionally, updating τ takes $O\left(1\right)$. Suppose the number of iterations is n, the total time complexity on the server side is

$$\begin{aligned} & 3\cdot O\left(N\cdot|C|\right)+O\left(1\right)+n\cdot\left[5\cdot O\left(N\cdot|C|\right)+O\left(1\right)\right] \\ = & (5\cdot n+3)\cdot O\left(N\cdot|C|\right)+(n+1)\cdot O\left(1\right) \end{aligned}. \tag{37}$$

On the worker side, there are four parts of computations from **Phase 2** and **Phase 3**. First, in **Phase 2**, each worker applies the groupings sent by the server to initially group the locations in his preference set, leading to $O\left(\rho\right)$ for him. Then, he invokes *OGAL* to adaptively refine his grouping based on Alg. 2 and Alg. 3. In particular, for Alg. 2, a worker first splits his grouping C, initializes and sorts the constructed structure set G, taking $O\left(1\right)$ and $O\left(\rho\right)+O\left(|C|\log|C|\right)$, as the sorting algorithm usually can be done in $O\left(|C|\log|C|\right)$. Then, he obtains the minimized the maximum sum (Lines 4-7), consuming $O\left(|C|^2\cdot\left(|C|-1\right)\right)$. Finally, he merges G to C, using $O\left(|C|\right)$. Since $\log x \approx x-1$ according to the *Taylor Expansion*, the total time complexity of Alg. 2 is $O\left(|C|+\rho\right)+O\left(|C|^3-|C|\right)+O\left(1\right)$ for each worker. For Alg. 3, each worker first calculates the elements in a group

and obtains the median, taking $O(\rho)$ and $O(|C|)$ respectively. Then, he merges each group, using $|C| \cdot O(1)$. Hence, the total time complexity of Alg. 3 is $|C| \cdot O(1) + O(|C|) + O(\rho)$ for each worker. At last, each worker conducts Alg. 4 in **Phase 3**. In particular, he first generates representative location using ρ locations and samples an obfuscated location for each group, leading to $O(\rho)$ and $O(|C|)$ respectively. Then, he computes and uploads the obfuscated distances to the server, which can be done in $2 \cdot O(\rho)$. Hence, the total time complexity of Alg. 4 is $3 \cdot O(\rho) + O(|C|)$ for each worker. Therefore, the total time complexity on the worker side is

$$M \cdot \left[O\left(|C|^3 + 2 \cdot |C|\right) + 6 \cdot O(\rho) + (|C| + 1) \cdot O(1)\right]. \quad (38)$$

From Eq. 37 and Eq. 38, we can see that the total time complexity is linear w.r.t. the number of tasks N or workers M. Moreover, the time consumptions of each worker and the server largely depend on the number of groups $|C|$ except for N and M. Furthermore, we have $|C| < \rho$, and we notice that the size of the preference set ρ is normally not unreasonably large due to crowdsourcing system overhead and economic budget. Besides, in practice, it is reasonable to assume that the server and each worker have strong computing power, and making use of distributed computing can further improve the runtime. As such, we believe that *CANOE* can provide an acceptable utility and efficiency trade-off for real-world deployments.

8 Experiments

In this Section, we first present the experimental setup. We then evaluate the performance of *CANOE* over two real-world datasets by varying different parameters. Finally, we verify the advantages of *CANOE*' key building blocks.

Datasets. In our experiments, we use two real-world datasets for evaluation, *T-Drive* [54] and *Tokyo* [55].

- **T-Drive:** It, *TD* for short, contains trajectories of more than 9,019 taxis and hundreds of thousands of passengers. Similar to [16], we assume that the drivers are SC workers and the passengers are SC tasks. We randomly sample 500 workers and 368 tasks.
- **Tokyo:** It, *TKY* for short, contains the locations of 325 subway stations and 503 offices in Tokyo. There are about 573,708 check-ins. Similar to [24], we assume that the office locations of users are the locations of workers and the locations of subway stations are the locations of tasks.

As for the preference set, as a common practice [56], we choose the first ρ tasks closest to workers as the preferred tasks of each worker. Meanwhile, we ensure that each task is within at least one worker's preference set.

Evaluation Metrics. We use the average travel distance (ATD) [14] to evaluate the performance of *CANOE*. In particular, we compute the real total Euclidean distance (in km) of the selected workers and their assigned tasks divided by the number of allocated tasks

$$ATD = \sum_{(w,t)} d(w,t) \Big/ |A|, \quad (39)$$

where A is the set of final task allocation (worker, task) pairs, and $d(w,t)$ is the Euclidean distance between the selected worker w and the assigned task t. The smaller, the better.

Competitors. For the competitors, as mentioned in related work, existing studies are designed specifically with the assumption that each worker can perform any tasks rather than one of a set of preferred tasks. Thus, for a fair comparison, we let all workers in the following competitors can only be assigned to one of a limited number of tasks and ensure that all the tasks are assigned.

- **NoPriv:** To show the utility loss due to privacy protection, we include a non-private version of *CANOE*, denoted by *NoPriv*. Both *NoPriv* and *CANOE* adopt the approach in [2] for task allocation.
- **DGO [14]:** It is a differential geo-obfuscation approach, which adds noise to each worker's location, namely differential geo-obfuscation.
- **IBA [24]:** It focuses on personalized privacy-preserving task allocation while adopting the laplace mechanism [20] with incentive that can allocate tasks effectively, namely incentive based task allocation.
- **PBA [16]:** It is a probability-based algorithm to assign tasks by adopting the laplacian mechanism, namely probability-based task allocation.
- **TBA [17]:** It uses a tree-based privacy mechanism to find the nearest reachable worker for each task, namely tree based task allocation.

The benefits for task preference have well demonstrated by the existing non-private task allocation approaches [3], [4], [56] while we focus on the privacy issue under this scenario. Therefore, we do not compare *CANOE* against the previous methods without preference, which are all orthogonal to our work.

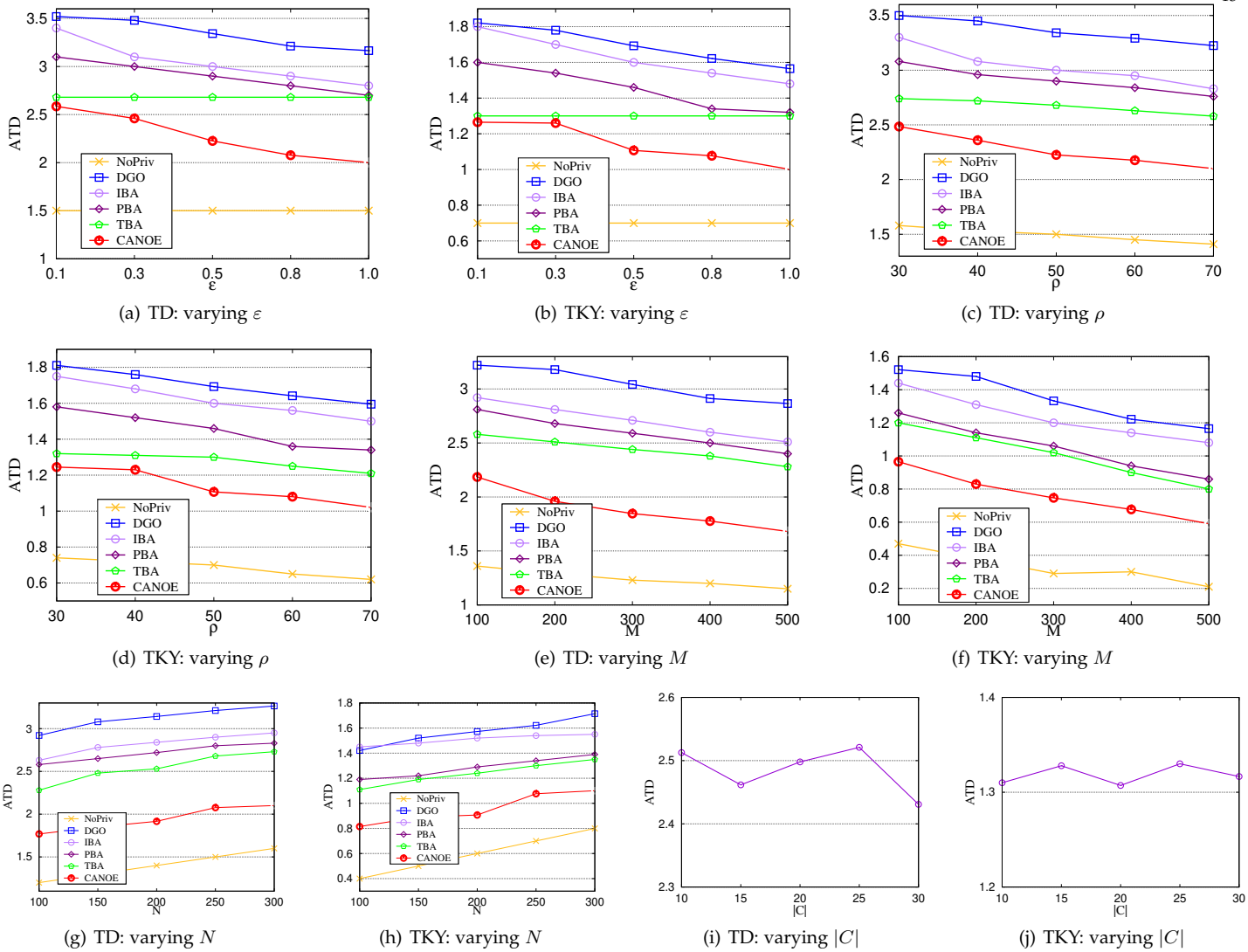
8.1 Performance Comparison

TABLE 2: Parameters that will affect *CANOE*

Notation	Values	Description
ε	0.1,**0.3**,0.5,0.8,1	Privacy budget
ρ	30, 40, **50**, 60, 70	Number of preferred tasks
M	100, 200, 300, **400**, 500	Number of workers
N	100, 150, 200, **250**, 300	Number of tasks
$\|C\|$	10, **15**, 20, 25,30	Number of groups

Five parameters will affect the performance of *CANOE*, which are shown in Tab. 2, where the values marked by bold font represent the corresponding default values.

Impact of ε. Figs. 5(a) and 5(b) show *ATD* of each competitor and *CANOE* when varying the privacy budget ε. We can see that *CANOE* outperforms all the competitors in all cases, and the relative superiority of *CANOE* is more emphasized while ε enlarges. The reasons can be explained as follows. On the one hand, for each group, by incorporating the tasks' locations in the same group to limit the range of the generated obfuscated location, it leads to a high chance of the obfuscated location falling into the nearby locations with its true location. On the other hand, through group-based noise addition, a task is more likely to be assigned to the optimal worker, which can minimize the average travel distance according to Theorem 9. These two jointly make *CANOE* be less sensitive to ε. In contrast, for *DGO*, the optimal worker assigned to each task may not prefer it. In such a case, the task can only be assigned to the nearest worker, resulting in larger *ATD*. For *IBA*, when the server decides which worker should be assigned to a task by judging the distance between this task and each worker, it conducts multiple probability comparisons for a certain worker in one iteration, which results in the calculated probability contains large error. Thus, it may lead to the wrong judgment, and unavoidably results in poor performance. For *PBA*, due to the randomness of probability approximation when post-processing the obfuscated locations generated by *PL*, they may



(a) TD: varying ε (b) TKY: varying ε (c) TD: varying ρ

(d) TKY: varying ρ (e) TD: varying M (f) TKY: varying M

(g) TD: varying N (h) TKY: varying N (i) TD: varying $|C|$ (j) TKY: varying $|C|$

Figure 5. Performance Comparison

be far away from the true locations. For *TBA*, it generates obfuscated locations based on the prior locations' scopes, while different scopes will exhibit different results, which makes it fail to be effectively applied for real-world deployments. Moreover, the locations' scopes may not be available in advance in practice.

Impact of ρ. Figs. 5(c) and 5(d) show *ATD* when varying the number of preferred tasks for each worker ρ, which denotes the number of elements in the preference set. We can observe that, with the increment of ρ, all the approach gets better performance. This is because on the premise of a fixed number of tasks, a larger ρ provides a higher possibility for each task to be assigned to a well-matched worker.

Impact of M. Figs. 5(e) and 5(f) show *ATD* when varying the number of workers M. We can observe that, with the increase of M, all the approaches get a better performance. This is because with more candidate workers, a task is more likely to be assigned to a proper worker, who really is close to this task. Thus, *ATD* will be smaller.

Impact of N. Figs. 5(g) and 5(h) show *ATD* when varying the number of tasks N. Generally, *ATD* becomes lower with the increment of N. This is because with more tasks while the number of workers remains unchanged, there are fewer choices for each task.

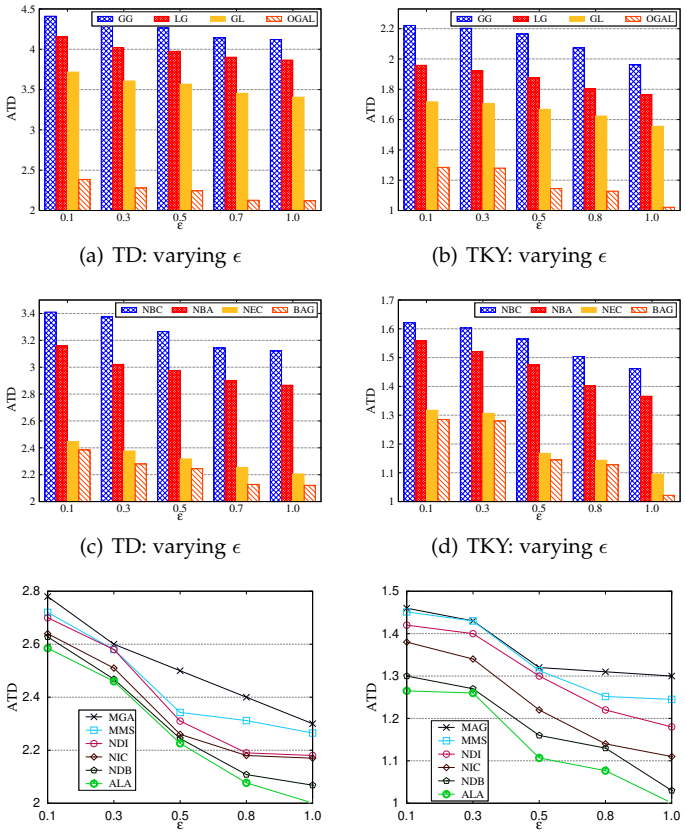
Impact of $|C|$. Figs. 5(i) and 5(j) show *ATD* when varying the number of groups $|C|$. In general, *ATD* remains stable when varying $|C|$. This is because, with local adjustment, we can adaptively determine the grouping for each worker, which makes our solution be more robust.

8.2 Effectiveness of *OGAL*

We do three parts of experiments to make a comprehensive and thorough evaluation on *OGAL*. We first verify the effectiveness of *OGAL* on the whole. Then, we verify the effectiveness of the building blocks *BAG* and *ALA*, which are used to conduct globally grouping and locally fine-tune each grouping respectively.

To verify the effectiveness of *OGAL*, we compare it with three baselines, which are *GG*, *LG* and *GL*. For *GG*, the server only groups the tasks, and each worker applies the grouping sent by the server to group the tasks in his preference set. For *LG*, each worker performs grouping locally. For *GL*, the server first groups the tasks using the existing KMeans clustering rather than our designed balance-aware clustering, and sends the grouping to each worker. Then, each worker conducts grouping locally using KMeans again instead of conducting local adjustment, with the received grouping from the server as the initialization.

Figs. 6(a) and 6(b) show the results. We have the following observations. First, *LG* performs better than *GG* when ε gradually enlarges. This is because the number of elements in the preference set is far less than the total number of tasks. Thus, *GG* will introduce too much information loss for a worker.



(a) TD: varying ϵ (b) TKY: varying ϵ

(c) TD: varying ϵ (d) TKY: varying ϵ

(e) TD: varying ϵ (f) TKY: varying ϵ

Figure 6. Effectiveness of *OGAL*

Second, *GL* performs better than *LG*, which demonstrates the effectiveness of group-based noise addition. Third, our *OGAL* significantly outperforms the baselines. The reason lies in that, we have better global grouping and local grouping, which can make the total noise be minimized.

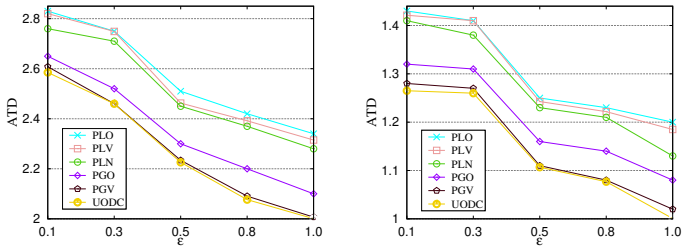
To verify the effectiveness of the building block *BAG*, we design three baselines, which are *NEC*, *NBA* and *NBC*. For *NEC*, it let $u_{ik} = 0$ if $u_{ik} < 0$ without our designed *Exponential Constraint*. For *NBA*, it feeds random values for S when solving *S-subproblem*, which aims to verify our alternative optimization scheme. For *NBC*, it conducts grouping without *Balance Constraint*, which aims to verify our noise analysis.

Figs. 6(c) and 6(d) show the results. We have the following observations. First, *NBA* and *NBC* lead to poor performance. This is because, without *Balance Constraint*, it may introduce too much noise for *NBC*. In addition, the formalized optimization problem may not converge for *NBA*. Second, *NEC* performs better than *NBA* and *NBC*, and performs worse than *BAG*. This verifies the effectiveness of minimizing noise and *Exponential Constraint*, as without *Exponential Constraint* it will divide the tasks that should be originally divided into multiple groups into the same group, which will introduce too much information loss. Third, *BAG* performs the best, as the total noise is minimized with convergence guarantee.

To verify the effectiveness of the building block *ALA*, we design five baselines, which are *NDB*, *NIC*, *NDI*, *MMS*, and *MGA*. For *NDB*, it does not conduct the operations from Theorem 5. For *NIC*, it does not tackle the *Intra-group Constraint* in Theorem 4. For *NDI*, it tackles the *Intra-group Constraint* without dynamic programming. For *MMS*, it only conducts *MGA* to determine the optimal number of tasks in each group. For *MGA*, it only conducts *MMS* to determine the optimal intra-group distances.

Figs. 6(e) and 6(f) show the results. We have the following observations. First, *MMS* performs better than *MGA*, and *NDB* performs better than the others except for *ALA*. This is because *MMS* paves the way for conducting *MGA*. Moreover, according to Theorem 9, the injected noise is less than *PL*. Since, the *Intra-group Constraint* is close related to *PL*, we can still get a better performance even without *MMS* to tackle the *Intra-group Constraint*. Second, *MMS* and *MGA* perform the worst. This demonstrates that we should tackle *MMS* and *MGA* jointly for local adjustment. This is because only performing one method cannot minimize the overall noise. Third, *ALA* performs the best. This is because by the dynamic programming based method *MMS* to get the optimal intra-group distances and the median based method *MGA* to get the optimal number of tasks in each group, we can minimize the total noise.

8.3 Effectiveness of *UODC*



(a) TD: varying ϵ (b) TKY: varying ϵ

Figure 7. Effectiveness of *UODC*

To verify the effectiveness of *UODC*, we design five baselines. Recall that *UODC* contains two phases, which are determining the representative location for each group and generating the obfuscated locations respectively. We design *PLO*, *PGO*, *PLV*, *PGV* and *PLN* by changing the representative location determination method and obfuscated location generation method. For *PLO*, we get the average representative location in each group and adopt *PL* to add noise. For *PGO*, we get the average representative location in each group and adopt the designed noise addition method in Eq. 27 to add noise. For *PLV*, we determine the representative location only considering that it stays as far away from the tasks in other groups as possible, and adopt *PL* to add noise. For *PGV*, the only difference from *PLV* is that we adopt the noisy location generation method in Eq. 27 to generate obfuscated locations. For *PLN*, we adopt the representative location determination method in Eq. 26 and *PL* to determine each representative location and generate obfuscated locations respectively.

Fig. 7 shows the results. We have the following observations. First, *PGO* and *PGV* perform better than the others except for *UODC*. This is because the designed obfuscated location generation method contributes more to the accuracy improvement. Second, *PGV* performs better than *PGO*, and *PLN* performs better than *PLV*, which demonstrate the effectiveness of the representative location determination method. Third, *UODC* performs the best. This is because the designed representative location determination method can make the information loss be minimized, and the designed obfuscated location generation method can make the injected noise be minimized.

9 Conclusion

In this paper, we present a geo-indistinguishable task allocation approach, called *CANOE*, which can provide desirable utility while still achieving rigorous Geo-I guarantee. We theoretically give its privacy, utility and complexity guarantees. In *CANOE*, two methods, optimized global grouping with adaptive lo-

cal adjustment method *OGAL*, and utility-aware obfuscated distance collection method *UODC*, are proposed to reduce the negative effect of adding noise. Experimental results with component-wise analyses validate the effectiveness of *CANOE*.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant No. 62072052 and No. 61872045), the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (Grant No. 61921003). Corresponding authors of this paper are Prof. Xiang Cheng and Prof. Sen Su.

References

[1] Y. Tong, Z. Zhou, Y. Zeng, L. Chen, and C. Shahabi, "Spatial crowdsourcing: a survey," *VLDB J.*, vol. 29, no. 1, pp. 217–250, 2020.

[2] Y. Zhao, J. Xia, G. Liu, H. Su, D. Lian, S. Shang, and K. Zheng, "Preference-aware task assignment in spatial crowdsourcing," in *Conf. on Associa. for the Advan. of Artific. Intelli., AAAI*, 2019, pp. 2629–2636.

[3] Y. Zhao, K. Zheng, H. Yin, G. Liu, J. Fang, and X. Zhou, "Preference-aware task assignment in spatial crowdsourcing: from individuals to groups," *IEEE Trans. on Knowl. and Data Engin.*, pp. 1–1, 2020.

[4] Y. Li, Y. Zhao, and K. Zheng, "Preference-aware group task assignment in spatial crowdsourcing: A mutual information-based approach," in *IEEE Conf. on Data Mining, ICDM*, J. Bailey, P. Miettinen, Y. S. Koh, D. Tao, and X. Wu, Eds. IEEE, 2021, pp. 350–359.

[5] P. Cheng, L. Chen, and J. Ye, "Cooperation-aware task assignment in spatial crowdsourcing," in *IEEE Conf. on Data Engine., ICDE*. IEEE, 2019, pp. 1442–1453.

[6] Y. Zhao, J. Guo, X. Chen, J. Hao, X. Zhou, and K. Zheng, "Coalition-based task assignment in spatial crowdsourcing," in *IEEE Conf. on Data Engine., ICDE*. IEEE, 2021, pp. 241–252.

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

[8] C. Chow, M. F. Mokbel, and X. Liu, "Spatial cloaking for anonymous location-based services in mobile peer-to-peer environments," *GeoInformatica*, vol. 15, no. 2, pp. 351–380, 2011.

[9] K. Fawaz and K. G. Shin, "Location privacy protection for smartphone users," in *ACM Conf. on Comput. and Commun. Secur., CCS*, 2014, pp. 239–250.

[10] K. Fawaz, H. Feng, and K. G. Shin, "Anatomization and protection of mobile apps' location privacy threats," in *Secur. Symp., SS*, 2015, pp. 753–768.

[11] C. Ma and C. W. Chen, "Nearby friend discovery with geo-indistinguishability to stalkers," in *Int. Conf. on Future Netw. and Commun.,FNC*, 2014, pp. 352–359.

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

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

[14] 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. of Conf. on World Wide Web, WWW*, 2017, pp. 627–636.

[15] L. Wang, G. Qin, D. Yang, X. Han, and X. Ma, "Geographic differential privacy for mobile crowd coverage maximization," in *Conf. on Associa. for the Advan. of Artific. Intelli., AAAI*, 2018, pp. 200–207.

[16] H. To, C. Shahabi, and L. Xiong, "Privacy-preserving online task assignment in spatial crowdsourcing with untrusted server," in *IEEE Conf. on Data Engine., ICDE*, 2018, pp. 833–844.

[17] Q. Tao, Y. Tong, Z. Zhou, Y. Shi, L. Chen, and K. Xu, "Differentially private online task assignment in spatial crowdsourcing: A tree-based approach," in *IEEE Conf. on Data Engine., ICDE*, 2020, pp. 517–528.

[18] N. Wang, X. Xiao, Y. Yang, J. Zhao, S. C. Hui, H. Shin, J. Shin, and G. Yu, "Collecting and analyzing multidimensional data with local differential privacy," in *IEEE Conf. on Data Engin., ICDE*, 2019, pp. 638–649.

[19] Y. Wang, L. Wu, and S. Wang, "A fully-decentralized consensus-based ADMM approach for DC-OPF with demand response," *IEEE Trans. Smart Grid*, vol. 8, no. 6, pp. 2637–2647, 2017.

[20] C. Dwork, F. McSherry, K. Nissim, and A. D. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography Conf., TCC*, 2006, pp. 265–284.

[21] X. Zhang, J. Ding, X. Li, T. Yang, J. Wang, and M. Pan, "Mobile crowdsensing task allocation optimization with differentially private location privacy," in *IEEE Conf. on Commun., ICC*, 2020, pp. 1–6.

[22] Y. Qian, Y. Ma, J. Chen, D. Wu, D. Tian, and K. Hwang, "Optimal location privacy preserving and service quality guaranteed task allocation in vehicle-based crowdsensing networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4367–4375, 2021.

[23] Y. Qian, Y. Jiang, M. S. Hossain, L. Hu, G. Muhammad, and S. U. Amin, "Privacy-preserving based task allocation with mobile edge clouds," *Inf. Sci.*, vol. 507, pp. 288–297, 2020.

[24] Z. Wang, J. Hu, R. Lv, J. Wei, Q. Wang, D. Yang, and H. Qi, "Personalized privacy-preserving task allocation for mobile crowdsensing," *IEEE Trans. Mob. Comput.*, vol. 18, no. 6, pp. 1330–1341, 2019.

[25] H. Shen, G. Bai, Y. Hu, and T. Wang, "P2TA: privacy-preserving task allocation for edge computing enhanced mobile crowdsensing," *J. Syst. Archit.*, vol. 97, pp. 130–141, 2019.

[26] W. Gong, B. Zhang, and C. Li, "Privacy-aware online task assignment framework for mobile crowdsensing," in *IEEE Conf. on Commun., ICC*, 2019, pp. 1–6.

[27] Q. Xu, Z. Su, M. Dai, and S. Yu, "APIS: privacy-preserving incentive for sensing task allocation in cloud and edge-cooperation mobile internet of things with SDN," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5892–5905, 2020.

[28] D. Zhang, Y. Ma, X. S. Hu, and D. Wang, "Toward privacy-aware task allocation in social sensing-based edge computing systems," *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11 384–11 400, 2020.

[29] S. Zakhary and A. Benslimane, "On location-privacy in opportunistic mobile networks, a survey," *J. Netw. Comput. Appl.*, vol. 103, pp. 157–170, 2018.

[30] J. Jiang, G. Han, H. Wang, and M. Guizani, "A survey on location privacy protection in wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 125, pp. 93–114, 2019.

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

[32] L. Yu, L. Liu, and C. Pu, "Dynamic differential location privacy with personalized error bounds," in *Annu. Netw. and Distributed Syst. Secur. Symp., NDSS*, 2017.

[33] S. Oya, C. Troncoso, and F. Pérez-González, "Back to the drawing board: Revisiting the design of optimal location privacy-preserving mechanisms," in *ACM Conf. on Comput. and Commun. Secur., CCS*, 2017, pp. 1959–1972.

[34] A. Pyrgelis, C. Troncoso, and E. D. Cristofaro, "What does the crowd say about you? evaluating aggregation-based location privacy," *PoPETs*, vol. 2017, no. 4, pp. 156–176, 2017.

[35] K. Chatzikokolakis, E. ElSalamouny, and C. Palamidessi, "Efficient utility improvement for location privacy," *PoPETs*, vol. 2017, no. 4, pp. 308–328, 2017.

[36] E. ElSalamouny and S. Gambs, "Optimal noise functions for location privacy on continuous regions," *Int. J. Inf. Sec.*, vol. 17, no. 6, pp. 613–630, 2018.

[37] S. Takagi, Y. Cao, Y. Asano, and M. Yoshikawa, "Geo-graph-indistinguishability: Protecting location privacy for LBS over road networks," in *Conf. on Data and Appli. Secur. and Privacy, DBSec*, 2019, pp. 143–163.

[38] S. Oya, C. Troncoso, and F. Pérez-González, "Is geo-indistinguishability what you are looking for?" in *Proc. of Workshop on Privacy in the Electronic Society,WPES*, 2017, pp. 137–140.

[39] F. McSherry, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis," in *ACM Conf. on Management of Data, SIGMOD*, 2009, pp. 19–30.

[40] Y. Yang, W. Liu, E. Wang, and J. Wu, "A prediction-based user selection framework for heterogeneous mobile crowdsensing," *IEEE Trans. Mob. Comput.*, vol. 18, no. 11, pp. 2460–2473, 2019.

[41] L. Pu, X. Chen, J. Xu, and X. Fu, "Crowdlet: Optimal worker recruitment for self-organized mobile crowdsourcing," in *IEEE Conf. on Com. Communi.*, 2016, pp. 1–9.

[42] E. Wang, Y. Yang, J. Wu, W. Liu, and X. Wang, "An efficient prediction-based user recruitment for mobile crowdsensing," *IEEE Trans. Mob. Comput.*, vol. 17, no. 1, pp. 16–28, 2018.

[43] G. Gao, J. Wu, M. Xiao, and G. Chen, "Combinatorial multi-armed bandit based unknown worker recruitment in heterogeneous crowdsensing," in *IEEE Conf. on Com. Communi.*, 2020, pp. 179–188.

[44] W. Liu, Y. Yang, E. Wang, and J. Wu, "Dynamic user recruitment with truthful pricing for mobile crowdsensing," in *IEEE Conf. on Com. Communi.*, 2020, pp. 1113–1122.

[45] G. Gao, H. Huang, M. Xiao, J. Wu, Y.-E. Sun, and Y. Du, "Budgeted unknown worker recruitment for heterogeneous crowdsensing using cmab," *IEEE Trans. Mob. Comput.*, 2021.

[46] M. J. Krieger, J.-B. Billeter, and L. Keller, "Ant-like task allocation and recruitment in cooperative robots," *Nature*, vol. 406, no. 6799, pp. 992–995, 2000.

[47] B. Guo, Y. Liu, L. Wang, V. O. K. Li, J. C. K. Lam, and Z. Yu, "Task allocation in spatial crowdsourcing: Current state and future directions," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1749–1764, 2018.

[48] J. Wang, F. Wang, Y. Wang, L. Wang, Z. Qiu, D. Zhang, B. Guo, and Q. Lv, "Hytasker: Hybrid task allocation in mobile crowd sensing," *IEEE Trans. Mob. Comput.*, vol. 19, no. 3, pp. 598–611, 2020.

[49] J. E. Spingarn, "Applications of the method of partial inverses to convex programming: Decomposition," *Math. Program.*, vol. 32, no. 2, pp. 199–223, 1985.
[50] K. S. Srivastava and S. R. Pattanaik, "Solving nonconvex feasibility problem on a sphere and a closed ball by douglas-rachford algorithm," *Asia Pac. J. Oper. Res.*, vol. 38, no. 1, pp. 2 050 042:1–2 050 042:20, 2021.
[51] N. Parikh and S. P. Boyd, "Block splitting for distributed optimization," *Math. Program. Comput.*, vol. 6, no. 1, pp. 77–102, 2014.
[52] H. Feng and S. Li, "A tracking differentiator based on taylor expansion," *Appl. Math. Lett.*, vol. 26, no. 7, pp. 735–740, 2013.
[53] R. Ghasemi, A. Nezakati, and M. R. Rabiei, "A strong law of large numbers for random sets in fuzzy banach space," *Adv. Fuzzy Syst.*, vol. 2020, pp. 8 185 061:1–8 185 061:10, 2020.
[54] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "Driving with knowledge from the physical world," in *ACM Conf. on Knowl. Disco. and Data Min., KDD*, 2011, pp. 316–324.
[55] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu, "Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns," *IEEE Trans. Syst., Man, and Cyber.: Syst.*, vol. 45, no. 1, pp. 129–142, 2015.
[56] B. Zhao, P. Xu, Y. Shi, Y. Tong, Z. Zhou, and Y. Zeng, "Preference-aware task assignment in on-demand taxi dispatching: An online stable matching approach," in *Conf. on Associa. for the Advan. of Artific. Intelli., AAAI*, 2019, pp. 2245–2252.

Pengfei Zhang is a Ph.D. Candidate from Beijing University of Posts and Telecommunications in China. His major is Computer Science. His current research interest focuses on privacy protection in mobile crowdsensing systems.

Xiang Cheng received the Ph.D. Degree in Computer Science from Beijing University of Posts and Telecommunications, China, in 2013. He is currently a Professor at the Beijing University of Posts and Telecommunications. His research interests include privacy-enhanced computing, data mining and knowledge engineering.

Sen Su received the Ph.D. degree in 1998 from the University of Electronic Science and Technology, China. He is currently a Professor at the Beijing University of Posts and Telecommunications. His research interests include data privacy, cloud computing and internet services.

Ning Wang is working toward the master's degree at the Beijing University of Posts and Telecommunications, China. His major is computer science. His research interests include data mining and data privacy.