

# Dynamic Private Task Assignment under Differential Privacy

Leilei Du

East China Normal University  
Shanghai, China  
leileidu@stu.ecnu.edu.cn

Peng Cheng

East China Normal University  
Shanghai, China  
pcheng@sei.ecnu.edu.cn

Libin Zheng

Sun Yat-sen University  
Guangzhou, China  
zhenglb6@mail.sysu.edu.cn

Wei Xi

Xi'an Jiaotong University  
Shaanxi, China  
xiwei@xjtu.edu.cn

Xuemin Lin

Shanghai Jiao Tong University  
Shanghai, China  
xuemin.lin@gmail.com

Wenjie Zhang

The University of New South Wales  
Sydney, Australia  
wenjie.zhang@unsw.edu.au

Jing Fang

East China Normal University  
Shanghai, China  
jingfang@stu.ecnu.edu.cn

**Abstract**—Data collection is indispensable for spatial crowdsourcing services, such as resource allocation, policy-making, and scientific explorations. However, privacy issues make it challenging for users to share their information unless receiving sufficient compensation. Differential privacy (DP) is a promising mechanism to release helpful information while protecting individuals' privacy. However, most DP mechanisms only consider a fixed compensation for each user's privacy loss. In this paper, we design a task assignment scheme that allows workers to dynamically improve their utility with dynamic distance privacy leakage. Specifically, we propose two solutions to improve the total utility of task assignment results, namely Private Utility Conflict-Elimination (PUCE) approach and Private Game Theory (PGT) approach, respectively. We prove that PUCE achieves higher utility than the state-of-the-art works. We demonstrate the efficiency and effectiveness of our PUCE and PGT approaches on both real and synthetic data sets compared with the recent distance-based approach, Private Distance Conflict-Elimination (PDCE). PUCE is always better than PDCE slightly. PGT is 50% to 63% faster than PDCE and can improve 16% utility on average when worker range is large enough.

**Index Terms**—Spatial Crowdsourcing, Differential Privacy

## I. INTRODUCTION

With the popularity of mobile computing, spatial crowdsourcing has emerged as a new paradigm for spatial task solutions involving human participation. Workers are appealed to share their data with servers in exchange for benefits. However, sometimes workers are reluctant to share due to vital privacy leakage (e.g., location), which can lead to extensive attacks such as identity theft, physical surveillance and stalking, and leakage of other sensitive information (e.g., individual health status, racial types, and religion views). For example, in ride-sharing, if a taxi driver submits his locations to the platform for task requests over a period of time (i.e., a month), a malicious platform attacker is able to guess the driver's range of activity and surveil him.

Differential privacy (DP) [1] is often used to protect individual data. It trades off utility and privacy by well designing the privacy budget ( $\epsilon$ ). However, different people have different demands for utility and privacy. For example, some confiden-

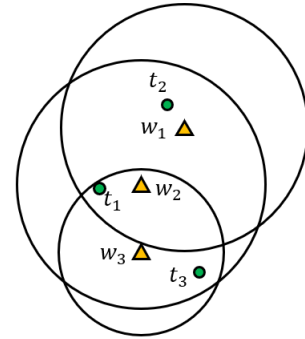


Fig. 1: Workers' locations with service areas and tasks' locations.

tial agencies pay great attention to privacy. They would rather get high-level privacy protection by sacrificing some utility. In ride-sharing, some taxi drivers would like to sacrifice some personal location privacy for higher incomes through serving more passengers. It is needed for different users to adjust their utility by altering the privacy protection level by themselves.

In this paper, we propose a dynamic private task assignment scheme such that workers can trade their location privacy for higher utilities. Consider the motivation example as follows:

**Example 1.** As shown in Figure 1, there are three workers:  $w_1$ ,  $w_2$  and  $w_3$ , and three tasks:  $t_1$ ,  $t_2$  and  $t_3$ . Each worker  $w_j$  competes for tasks with smaller distances. However, in order to protect their locations, all workers employ a differential privacy mechanism to obfuscate their distances to tasks, and send the obfuscated distances to the server. Assume the server will assign workers to tasks based on their reported obfuscated distances to minimize the total distance as:  $\{\langle t_1, w_3 \rangle, \langle t_2, w_1 \rangle, \langle t_3, w_2 \rangle\}$ . We also assume that the server is untrusted in this example, which means the obfuscated distances on the server can be accessed by workers if they want. Then, worker  $w_3$  can sacrifice some his location privacy to report a closer obfuscated distance with task  $t_3$ . The updated assignment will be  $\{\langle t_1, w_2 \rangle, \langle t_2, w_1 \rangle, \langle t_3, w_3 \rangle\}$  with a smaller total distance.

In this paper, we study the privacy-aware task assignment (PA-TA) problem in spatial crowdsourcing, where workers

can dynamically adjust their privacy protection levels for higher utilities. Specifically, we assume a privacy setting where spatial crowdsourcing workers are curious and want to protect their location privacy thus only report obfuscated distances to the server during the task assignment phase without relying on a trusted server. Similar to the existing location protection studies in spatial crowdsourcing [2], [3], [4], we assume the server is untrusted, and thus cannot guarantee the security of received obfuscated distances, which means other entities (e.g., curious workers) can access to the obfuscated distances from the server. In this paper, we handle the task assignment in a multi-proposal enabled batch-based style, where in a given time window each worker can propose to an available task for multiply times with different obfuscated distances to improve their utilities until the end of the time window. We first formally define the PA-TA problem. To improve the accuracy of comparing obfuscated distances, we propose a new comparison method, *Partial Probability Comparison Function* (PPCF), which can resolve the comparison between a real distance and an obfuscated distance. We prove our PPCF is better than the existing method, *Probability Compare Function* (PCF) [3], in both theory and practice. To solve PA-TA, we propose two solutions, namely Private Utility Conflict-Elimination (PUCE) and Private Game Theoretic Approach (PGT). PUCE is a greedy based algorithm. PGT is on a game-theoretic approach and can achieve higher accuracy than PUCE when the worker range is larger than 1.4 on synthetic data sets. The contributions of this paper are as follows.

(1) We formally define the privacy-aware task assignment problem to supports dynamic privacy budget adjustment for workers in spatial crowdsourcing in Section III.

(2) We propose a greedy-based algorithm, namely Private Utility Conflict-Elimination (PUCE), in Section V, and a game-theoretic approach, namely Private Game Theoretic Approach (PGT), in Section VI.

(3) We test our approaches on both synthetic and real data sets to show their efficiency and effectiveness in Section VII.

## II. RELATED WORK

**Task Assignment in Spatial Crowdsourcing.** Most task assignments in spatial crowdsourcing target on maximizing total utility. Deng et al. [5] define the total utility as the total number of performed tasks. Zhang et al. [6] maximize the total acceptance ratio of workers. Zhao et al. [7] propose algorithms to maximize the total rewards of the assigned tasks. Tong et al. [8] and Wang et al. [9] maximize the total expected rewards of the assigned tasks. The conventional methods to achieving optimized total utility are exact methods [10], [11] and greedy methods [12], [13], [14]. In order to achieve higher utilities, game theoretic methods for task assignment are proposed recently. Ni et al. [15] declare that the tasks may have some dependencies among them and give the definition of dependency-aware spatial crowdsourcing (DA-SC). They propose a game-theoretic approach to solve DA-SC, and show that the game-theoretic approach is better than the greedy algorithms through the experiment. Zhao et al. [16]

focus on the problem of Fairness-aware Task Assignment, which is to minimize the payoff difference among workers and to maximize the average worker payoff. They model the problem as a multi-player game and propose two game-theoretic methods.

**Privacy Protection in Spatial Crowdsourcing.** Differential privacy [1] is a golden tool for privacy protection and private data release. According to the existing of the trusted entity, it can be classified into two categories: 1) Central Differential Privacy (CDP) [17]; 2) Geo-Indistinguishability (Geo-I) [18] and Local Differential Privacy (LDP) [19].

To et al. [20] adopt *Private Spatial Decomposition* (PSD) [21] to create obfuscated data releases of workers and devise a geocast mechanism for task request dissemination to protect the privacy of workers' locations. However, it needs a trusted entity to help sanitize workers' location data. Wang et al. [22] study *Bayesian attack* [18], [23] on sparse mobile crowdsourcing and propose a privacy-preserving framework to reduce the data quality loss caused by differential location obfuscation. They provide the method to get the optimal location obfuscation matrix satisfying  $\epsilon$ -differential privacy. It can be used to protect workers' location without relying on the trust entity. To et al. [2] propose a privacy-aware framework that protects the privacy of both tasks and workers in spatial crowdsourcing without any trusted entity. It employs Geo-I to transform both tasks' and workers' locations into obfuscated locations. The platform can identify a set of candidate workers for the task requester through these obfuscated locations without knowing the real locations of both workers and the task. Wang et al. [3] also assume that no trusted entity exist, but they propose a method that achieves local differential privacy. These works get rid of reliance on trusted third parties. However, they only protect individual privacy without inspiring tasks or workers to participate in the platform.

**Private Data Compensation.** In order to motivate requesters and workers to join spatial crowdsourcing platforms while protecting their location privacy, we need a connection between their utility and privacy cost. Jin and Zhang [24] provide a framework for spectrum-sensing participants selection, which achieves differential location privacy, approximate social cost minimization, and truthfulness simultaneously. Ghosh et al. [25] model the utility of competing agents considering privacy cost. They hold the privacy cost related to some unknown quantities  $v$  and suppose the privacy cost is changing linearly with privacy budget  $\epsilon$  ( $\epsilon v$ ). Nissim et al. [26] argue that  $\epsilon v$  should be the upper bound rather than the total privacy cost. They propose a privacy-aware mechanism with  $v$  below a certain threshold. Xiao [27] proposes two models for quantifying an agent's privacy cost using mutual information and max divergence, respectively. However, it requires the privacy variable  $\delta > 0$ . Wang et al. [3] propose a personalized privacy-preserving task allocation method. They define *Probability Compare Function* (PCF) to compare two noise values with the acknowledgment of their privacy budget. Besides, they propose *Probabilistic Winner Selection Mechanism* to minimize the total travel distance and *Vickrey Payment Determination*

TABLE I: Notations.

Variable	Description
$t_i$	the $i$ -th task
$w_j$	the $j$ -th worker
$d_{i,j}$	the real distance from $t_i$ to $w_j$
$\hat{d}_{i,j}$	the obfuscated distance from $t_i$ to $w_j$
$\tilde{d}_{i,j}$	the effective obfuscated distance from $t_i$ to $w_j$
$\epsilon_{i,j}$	the privacy budget vector owned by $w_j$ to propose to $t_i$
$\epsilon_{i,j}^{(u)}$	the $u$ -th element in $\epsilon_{i,j}$
$\tilde{\epsilon}_{i,j}$	the effective privacy budget
$\mathbf{b}_{i,j}$	the state vector corresponding to $\epsilon_{i,j}$
$b_{i,j}^{(u)}$	the $u$ -th element in $\mathbf{b}_{i,j}$ recording whether $\epsilon_{i,j}^{(u)}$ has been used
$s_{i,j}$	the state recording whether $t_i$ matches $w_j$

*Mechanism* to determine the appropriate payment to each winner of workers satisfying truthfulness, profitability, and probabilistic individual rationality. However, all workers can only have a fixed budget for each task and cannot dynamically compete for tasks with higher utilities.

### III. PROBLEM DEFINITION

**Definition 1** (Spatial Tasks). Let  $t_i$  denote a task. Its location and value are denoted as  $l_i$  and  $v_i$ , respectively.

Here,  $v_i$  is an inherent property of  $t_i$ , and a worker will gain  $v_i$  revenue if he serves  $t_i$ .

**Definition 2** (Spatial Workers). Let  $w_j$  denote a worker located at  $l_j$ . His service area is denoted as  $A_j$  with a service radius  $r_j$ .

$A_j$  is a circle area centered at  $l_j$  with radius  $r_j$  (also called *worker range* in the experiment). Let set  $R_j$  denote all tasks in  $A_j$ .  $w_j$  only proposes to those tasks in  $R_j$ .

To make the distance and the privacy budget comparable with the task value, we define the Distance Value Function ( $f_d$ ) in Definition 3 and Privacy Budget Function ( $f_p$ ) in Definition 4 to unify the measurement.

**Definition 3** (Distance Value Function,  $f_d$ ). Given a distance  $d \in R^*$ , a function  $f_d : R^* \rightarrow R^*$  is called *distance value function*, which takes  $d$  as the input and outputs a value  $v$ . It satisfies that  $f_d(0) = 0$ ,  $f_d(\cdot) \geq 0$ .

**Definition 4** (Privacy Budget Value Function,  $f_p$ ). Given a privacy budget  $\epsilon \in R^*$ ,  $f_p : R^* \rightarrow R^*$  is a *privacy budget value function*, which takes  $\epsilon$  as input and outputs a value  $v$ . It satisfies that  $f_p(0) = 0$ ,  $f_p(\cdot) \geq 0$  and  $\forall \epsilon_1, \epsilon_2 \in R$ ,  $f_p(\epsilon_1) + f_p(\epsilon_2) = f_p(\epsilon_1 + \epsilon_2)$ .

$f_d$  transforms a distance value into a task value.  $f_p$  transforms a privacy budget value into a task value.  $f_d$  and  $f_p$  are defined as monotone increasing functions and  $f_d(0) = f_p(0) = 0$ . Besides,  $f_p$  is a linear function in this paper and we will consider other types of functions in the future work.

**Definition 5** (Privacy-aware Task Assignment Problem). Given a set of tasks  $\mathcal{T}$ , a set of workers  $\mathcal{W}$ , and a set of obfuscated worker-and-task distances  $\{\hat{d}_{i,j} | i \in [m], j \in [n]\}$ , where each  $\hat{d}_{i,j}$  is added with a noise  $\eta_{i,j}$  subjecting to distribution  $DF(\epsilon_{i,j})$ , a PA-TA problem is to find a match  $M$  between workers and tasks subject to the working area constraint of workers, such that

$$\begin{aligned}
& \max \sum_{t_i \in \mathcal{T}} \sum_{w_j \in \mathcal{W}} (s_{i,j} \cdot (v_i - f_d(d_{i,j})) - f_p(\mathbf{b}_{i,j} \cdot \epsilon_{i,j})) \\
& s.t. \sum_{t_i \in \mathcal{T}} s_{i,j} \leq 1, \quad \forall i = 1, 2, \dots, m \\
& \sum_{w_j \in \mathcal{W}} s_{i,j} \leq 1, \quad \forall j = 1, 2, \dots, n \\
& \sum_{z \in Z} b_{i,j}^{(z)} \leq Z, \quad \forall z = 1, 2, \dots, Z \\
& s_{i,j}, b_{i,j} \in \{0, 1\}, \quad \forall i = 1, 2, \dots, m; \forall j = 1, 2, \dots, n
\end{aligned}$$

where  $s_{i,j}$  is the matching state representing whether task  $t_i$  is allocated to worker  $w_j$ .  $s_{i,j} = 1$ , if  $t_i$  is allocated to  $w_j$ ; otherwise,  $s_{i,j} = 0$ .  $v_i$  is the value of task  $t_i$ .  $f_d$  is a Distance Value Function transforming distance to value cost.  $f_p$  is a Privacy Budget Value Function transforming privacy cost to value cost.  $\epsilon_{i,j} = \langle \epsilon_{i,j}^{(1)}, \dots, \epsilon_{i,j}^{(Z)} \rangle$  is the privacy budget vector between task  $t_i$  and worker  $w_j$ , where  $\epsilon_{i,j}^{(u)}$  ( $u \in Z$ ) stands for the  $u$ -th proposal of worker  $w_j$  to task  $t_i$ .  $\mathbf{b}_{i,j} = \langle b_{i,j}^{(1)}, \dots, b_{i,j}^{(Z)} \rangle$  is the state vector corresponding to  $\epsilon_{i,j}$ . Take  $\mathbf{b}_{1,2} = \langle 1, 1, 0, 0, 0 \rangle$  as an example. It means in the total competition,  $w_2$  can propose to  $t_1$  five times and has already proposed twice with the privacy leakage  $\epsilon_{1,2}^{(1)}$  and  $\epsilon_{1,2}^{(2)}$ .

The objective of PA-TA is to find a one-to-one match that maximizes the total profit on the platform. In the objective function, there are three important parts to construct the matching profit between  $t_i$  and  $w_j$ : task value  $v_i$ , distance value cost  $f_d(d_{i,j})$  and privacy cost  $f_p(\mathbf{b}_{i,j} \cdot \epsilon_{i,j})$ . We model the matching profit as the linear combination of the three parts. Note that, the privacy cost is concerned for the process of “ $w_j$  proposing to  $t_i$ ” but not for the final matching state. Thus,  $f_p(\mathbf{b}_{i,j} \cdot \epsilon_{i,j})$  is not affected by  $s_{i,j}$ .

We give some of the frequently used variables in Table I

### IV. REVIEW OF CONFLICT ELIMINATION ALGORITHM

Conflict Elimination Algorithm (CEA) [3] is a related work that can resolve the winner conflict problem and can be used as a subroutine in our proposed algorithm, thus we first quickly review CEA. Here, workers are regarded as competitors. When there are more than one worker competing for one task, there will be a conflict, called winner conflict. The problem of resolving all these conflicts is called winner conflict problem.

Given all distances from each task-worker pair, CEA constructs the distance rank matrix  $A_{m \times n} = (a_{i,k})_{m \times n}$  where  $a_{i,k}$  stands for the index of the worker who is the  $k$ -th nearest from  $t_i$ . For example,  $a_{i,k} = j$  means  $w_j$  is the  $k$ -th nearest worker of  $t_i$ .

For any conflict worker  $w_c$  selected by  $\varphi$  tasks, CEA allocates only one task to  $w_c$  and finds another candidate other than  $w_c$  for each of the rest  $\varphi - 1$  conflict tasks. Thus, for each conflict worker  $w_c$ , there will be  $\varphi$  candidate distance choices as shown in equation 1:

$$\begin{cases}
C_1 : D_{c_1} = D(a_{c_1,1}) + D(a_{c_2,2}) + \dots + D(a_{c_\varphi,\varphi}) \\
C_2 : D_{c_2} = D(a_{c_1,2}) + D(a_{c_2,1}) + \dots + D(a_{c_\varphi,\varphi}) \\
\vdots \\
C_\varphi : D_{c_\varphi} = D(a_{c_1,\varphi}) + D(a_{c_2,2}) + \dots + D(a_{c_\varphi,1})
\end{cases} \quad (1)$$

where  $c_u$  ( $1 \leq u \leq \varphi$ ) stands for the  $u$ -th solution: allocating  $w_c$  to  $t_{c_u}$  and other tasks are allocated to the successive workers.

TABLE II: Distance rank matrix.

Task/Rank	1	2	3
$t_1$	$w_1$ (9.06)	$w_2$ (9.85)	$w_3$ (12.04)
$t_2$	$w_3$ (2.09)	$w_1$ (10.44)	$w_2$ (12.59)
$t_3$	$w_3$ (2.00)	$w_2$ (11.28)	$w_1$ (18.87)

To choose the best solution from  $\varphi$  choices in equation 1, we need to compare four distance values. For example, to compare  $\mathcal{C}_u$  and  $\mathcal{C}_v$  ( $1 \leq u, v \leq \varphi$ ), we need to compare  $D(a_{c_u,1}) + D(a_{c_v,2})$  with  $D(a_{c_v,1}) + D(a_{c_u,2})$ . If the distances are obfuscated distances, we have to compare four Laplace random variables. In CEA, it supposes that the difference between the travel distances for different tasks is relatively small for the same worker (i.e.,  $D(a_{c_u,1}) \simeq D(a_{c_v,1})$ ). Then, CEA only needs to compare two Laplace random variables, which can be calculated by *Probability Compare Function* [3].

**Definition 6** (Probability Compare Function [3]). Given two values  $d_a$  and  $d_b$  with their obfuscated values  $\hat{d}_a = d_a + \text{Lap}(x, 1/\epsilon_a)$  and  $\hat{d}_b = d_b + \text{Lap}(x, 1/\epsilon_b)$  ( $\text{Lap}(x, y)$  is a random variable drawn from Laplace distribution with parameter  $x, y$ ), a function  $f : \mathbb{R}^4 \rightarrow [0, 1]$  is called a probability compare function (PCF) if  $\text{PCF}(\hat{d}_a, \hat{d}_b, \epsilon_a, \epsilon_b) = \Pr[d_a < d_b]$ .

For Example, suppose there are 3 tasks and 3 workers, their distance rank matrix is shown in Table II. Each element in the table stands for the worker and his relative distance to the corresponding task. For  $w_3$ , both  $t_2$  and  $t_3$  will choose him at the first rank, thus  $w_3$  is a conflict worker. We have  $\mathcal{C}_1 : D_2 = D(a_{2,1}) + D(a_{3,2})$  and  $\mathcal{C}_2 : D_3 = D(a_{2,2}) + D(a_{3,1})$ . To make a choice between  $\mathcal{C}_1$  and  $\mathcal{C}_2$  (choose the minimal one), it supposes  $D(a_{2,1}) \simeq D(a_{3,1})$ , and thus only needs to compare  $D(a_{3,2})$  with  $D(a_{2,2})$ . Since  $D(a_{2,2}) < D(a_{3,2})$ ,  $\mathcal{C}_2$  is selected.

## V. PRIVATE UTILITY CONFLICT-ELIMINATION (PUCE)

A direct method to solve our matching problem is collecting all workers' proposals to tasks with privacy budgets and obfuscated distances and using the Hungarian algorithm to get the optimal matching. Here, the Hungarian matching algorithm [28], also called the Kuhn-Munkres algorithm, is one classical method to exactly solve maximum bipartite matching problem with the time complexity of  $O(n^3)$ , where  $n$  is the number of vertices in either part of the bipartite graph. However, to use the Hungarian algorithm, we have to compare the path length calculated by summing many obfuscated distances, which needs complex comparisons and has low accuracy. In this section, we propose a private utility conflict-elimination (PUCE) algorithm to solve PA-TA problem. Due to each worker can propose to multiple tasks in each round, PUCE greedily chooses the worker-and-task pair that maximizes the subjective function of PA-TA.

### A. Comparison and Estimation of Obfuscated Distances

Before introduce PUCE algorithm, we first explain three necessary techniques on: 1) how to calculate a suitable obfuscated distance when there is a series of obfuscated distances for a given task and a given worker; 2) how to compare a real

distance with an obfuscated distance; 3) how to compare two utilities when knowing the obfuscated distances.

In this paper, according to the objective function of PA-TA, we define the utility of worker  $w_j$  conducting task  $t_i$  as:

$$U_j(i) = v_i - f_d(d_{i,j}) - \sum_{t_i \in \mathcal{T}} f_p(\mathbf{b}_{i,j} \cdot \boldsymbol{\epsilon}_{i,j}) \quad (2)$$

**Effective Obfuscated Distance and Effective Privacy Budget.** In the process of our task assignment,  $w_j$  may propose to  $t_i$  many times, which means  $w_j$  will submit more than one obfuscated distance  $\hat{d}_{i,j}$  to the server. For the server, it needs to determine an obfuscated distance (we call it *effective obfuscated distance*) for  $d_{i,j}$  to make comparison. For other workers, they also need the *effective obfuscated distance* to compare with the distances of themselves. Thus, we need a method to calculate the *effective obfuscated distance* in a series of obfuscated distances and ensure the *effective obfuscated distance* supports comparison (i.e., supporting PCF).

We first adopt *maximum likelihood estimation* (MLE) [29] to get a distance interval  $\tilde{d}$  from a worker  $w$ 's release set  $\mathbf{DE} = \{(\hat{d}_1, \epsilon_1), (\hat{d}_2, \epsilon_2), \dots, (\hat{d}_u, \epsilon_u)\}$  for a task  $t$ . Let  $\mathbf{DE}.\hat{\mathbf{d}}$  denote the set  $\{\hat{d}_1, \hat{d}_2, \dots, \hat{d}_u\}$  in  $\mathbf{DE}$ . Let  $L(X) = L(\hat{d}_1, \hat{d}_2, \dots, \hat{d}_u; X) = \prod_{k=1}^u \Pr[\hat{d}_k; X]$ , where  $\Pr[\hat{d}_k; X]$  is the probability function of  $\text{Lap}(\epsilon_k)$ . When the server gets  $\mathbf{DE}$ , it calculates the estimation of  $d$  as follows.

$$\begin{aligned} \tilde{d} &= \arg \max_d \prod_{k=1}^u \frac{\epsilon_k}{2} \exp(-|\hat{d}_k - d| \cdot \epsilon_k) \\ &= \arg \min_d \sum_{k=1}^u \epsilon_k \cdot |\hat{d}_k - d|. \end{aligned}$$

The value of  $\tilde{d}$  is all points on a line segment. We limit the domain of  $d$  in  $\mathbf{DE}.\hat{\mathbf{d}}$  to get the only estimation of  $d$  (supporting comparison). This estimation of  $d$  is the *effective obfuscated distance* and we denote it as  $\tilde{d}$ . We call the corresponding privacy budget (denoted by  $\tilde{\epsilon}$ ) of  $\tilde{d}$  in the pair as *effective privacy budget* and call the pair  $(\tilde{d}, \tilde{\epsilon})$  as *effective distance-budget pair*.

For example, suppose  $w_1$  releases 3 pairs of obfuscated distance and privacy budget to  $t_1$ :  $\mathbf{DE} = \{(0.1, 0.2), (0.2, 0.9), (0.3, 0.1)\}$ . Then we can calculate the effective distance-budget pair as  $(\tilde{d} = 0.2, \tilde{\epsilon} = 0.9)$ .

**Partial Probability Compare Function (PPCF).** If  $w_{j_1}$  want to compare his distance from himself to  $t_i$  with the effective obfuscated distance  $\hat{d}_{i,j_2}$  of  $w_{j_2}$  to  $t_1$ ,  $w_{j_1}$  can utilize the real distance  $d_{i,j_1}$  instead of  $\hat{d}_{i,j_1}$  or  $\tilde{d}_{i,j_1}$  to achieve a more accurate comparison result. Thus we need a method for the comparison between a real distance and an obfuscated distance. Suppose there are two values  $d_i$  and  $d_j$ . The obfuscated value of  $d_j$  is  $\hat{d}_j$ , which is calculated by adding noise  $\eta_j$  drawn from a type of distribution  $DF(\epsilon_j)$ . Then, we have

$$\begin{aligned} \hat{d}_j &= d_j + \eta_j, \quad \eta_j \sim DF(\epsilon_j), \\ \Pr[d_i < d_j] &= \Pr[d_i < \hat{d}_j - \eta_j] \\ &= \Pr[\eta_j < \hat{d}_j - d_i]. \end{aligned}$$

Let  $f(x)$  be the probability density function of  $\eta_j$ , then

$$\Pr[d_i < d_j] = \int_{-\infty}^{\hat{d}_j - d_i} f(\eta_j) d\eta_j.$$



---

**Algorithm 1: WorkerProposal**


---

**Input:** Not winning worker set  $NWW$   
**Output:** Candidate list  $CL$

```

1 Initialize candidate list  $CL$  as  $m$  empty sets;
2 for each worker  $w_j$  in  $NWW$  do
3   Initialize applicable list  $BL$  as empty list;
4   for each task  $t_i$  in  $R_j$  do
5     if  $w_j$ 's privacy budget has been exhausted then
6        $\quad$  continue;
7      $U_j(i) = v_i - f_d(d_{i,j}) - \sum_{t_i \in T} f_p(b_{i,j} \cdot \epsilon_{i,j})$ ;
8     if  $U_j(i) \leq 0$  then
9        $\quad$  continue;
10    Get  $(\tilde{d}_{i,win(i)}, \tilde{\epsilon}_{i,win(i)})$  of  $w_{win(i)}$  from the server;
11    Calculate new  $(\hat{d}_{i,j}, \hat{\epsilon}_{i,j})$ ;
12    Calculate  $\hat{d}'_{i,win(i),j}$  by Equation 4;
13    if  $PPCF(d_{i,j}, \hat{d}'_{i,win(i),j}, \epsilon_{i,win(i)}) \leq 0.5$  then
14       $\quad$  continue;
15    if  $PCF(\hat{d}_{i,j}, \hat{d}'_{i,win(i),j}, \tilde{\epsilon}_{i,win(i)}, \tilde{\epsilon}_{i,j}) \leq 0.5$  then
16       $\quad$  continue;
17    Add  $\hat{d}_{i,j}$  to  $CL[i]$ 
18 return  $CL$ ;
```

---

Similar to PCF, we define  $PPCF(d_i, \hat{d}_j, \epsilon_j) = \Pr[d_i < d_j]$ . If the distribution of  $DF(\epsilon_j)$  is symmetric about the y axis (e.g., Laplace distribution), then

$$PPCF(d_i, \hat{d}_j, \epsilon_j) > \frac{1}{2} \Leftrightarrow d_i < \hat{d}_j. \quad (3)$$

Our PPCF is better than PCF as shown in Theorem V.1. Please refer to the details of the proof in Appendix IX-A in our technical report [30].

**Theorem V.1.** For any given distance  $d_x, d_y, \epsilon_x, \epsilon_y$  satisfying  $d_x < d_y$ . Let  $\eta_x \sim Lap(0, 1/\epsilon_x), \eta_y \sim Lap(0, 1/\epsilon_y)$ . Let  $\hat{d}_x = d_x + \eta_x, \hat{d}_y = d_y + \eta_y$ . Then  $\Pr[PCF(\hat{d}_x, \hat{d}_y, \epsilon_x, \epsilon_y) > \frac{1}{2}] \leq \Pr[PPCF(d_x, d_y, \epsilon_y) > \frac{1}{2}]$ .

**Comparison Transformation from Utility to Distance.** After receiving proposals of workers, the server needs to eliminate conflict among workers for each task. We can easily use CEA directly to choose only one worker for each task. However, in CEA, the comparison is based on obfuscated distances rather than utility functions, which does not satisfy our optimized goal. If we use the utility directly as the comparison object, the server will know the utility value in each round, which leaks the real distance between tasks and workers.

In order to handle the problem above, we convert the utility comparison into the distance comparison and then use CEA to choose the high-utility one under the distance form. For any two workers  $w_a$  and  $w_b$ , they hold tasks  $t_x$  and  $t_y$ , respectively. Their utilities are  $U_a(x)$  and  $U_b(y)$ , respectively. Let  $V_a(x) = U_a(x) + f_d(d_{x,a})$  and  $V_b(y) = U_b(y) + f_d(d_{y,b})$ . Then we have

$$\begin{aligned} \Pr(U_a(x) > U_b(y)) &= \Pr(V_a(x) - f_d(d_{x,a}) > V_b(y) - f_d(d_{y,b})) \\ &= \Pr(f_d^{-1}(V_a(x)) - d_{x,a} > f_d^{-1}(V_b(y)) - d_{y,b}) \\ &= \Pr(d_{x,a} < d_{y,b} + f_d^{-1}(V_a(x)) - f_d^{-1}(V_b(y))). \end{aligned}$$

$$\text{Let } \hat{d}'_{y,b,a} = \hat{d}_{y,b} + f_d^{-1}(V_a(x)) - f_d^{-1}(V_b(y)), \quad (4)$$

$$\text{thus, } \Pr(U_a(x) > U_b(y)) = \Pr(\eta_{x,a} - \eta_{y,b} > \hat{d}_{x,a} - \hat{d}'_{y,b,a}) \\ = PCF(\hat{d}_{x,a}, \hat{d}'_{y,b,a}, \epsilon_{x,a}, \epsilon_{y,b}).$$

Therefore, we can calculate  $\hat{d}'_{y,b,a}$  for each pair of  $w_a$  and  $w_b$  with the same task  $t_y$  and use PCF function to compare the

---

**Algorithm 2: WinnerChosen**


---

**Input:** Candidate list  $CL$ , last term allocation list  $AL'$   
**Output:** Allocation list  $AL$ , updating state  $upd$

```

1 if All set in  $CL$  are empty then
2    $\quad$  return  $(AL', false)$ 
3 Initialize each update task set list  $TSL$  as empty list;
4 Initialize  $AL$  as  $m$  null values;
5 Initialize competing table  $CT$  as empty table;
6 for Each candidate set  $CS_i$  in  $CL$  do
7   if  $CS_i$  is empty then
8      $\quad$  Set  $AL[i] = AL'[i]$ ;
9   else
10    Add  $CT[i] = CS_i \cup \{\hat{d}_{i,win(i)}\}$  to  $CT$ ;
11    Calculate  $\hat{d}'_{i,a,b}$  for each pair in  $CT[i]$ ;
12    Sort  $CT[i]$  in descending order by  $PCF(\hat{d}'_{i,a,b}, \hat{d}_{i,b}, \epsilon_{i,a}, \epsilon_{i,b})$ ;
13 Get updated matching  $M$  set by using CEA for  $CT$ ;
14 Add  $M$  to  $AL$ ;
15 return  $(AL, true)$ ;
```

---

utility. Similarly, we can compare  $U_a(x)$  and  $U_b(y)$  through PPCF:

$$\begin{aligned} \Pr(U_a(x) > U_b(y)) &= \Pr(\eta_{y,b} < \hat{d}'_{y,b,a} - d_{x,a}) \\ &= PPCF(d_{x,a}, \hat{d}'_{y,b,a}, \epsilon_{y,b}). \end{aligned}$$

### B. The PUCE Algorithm

We suppose that  $w_j$  will propose to all tasks  $R_j$  within area  $A_j$ . In order to further decline unnecessary privacy costs, we add an extra judgment for workers through the PPCF function.

The worker proposal process and winner-chosen algorithm are respectively shown in Algorithm 1 and Algorithm 2.

In Algorithm 1, each worker  $w_j$  checks all the tasks in his service area and judges whether it is worth to complete for the tasks (check whether  $U_j(i) > 0$  for  $t_i \in R_j$ ). Besides, he also judges whether he has advantages over the before-winner worker for these tasks by utility comparison. The utility comparison is shown from line 10 to line 16. If the two conditions are satisfied,  $w_j$  will propose to this task with a new privacy budget and obfuscated distance.

Algorithm 2 takes candidate list  $CL$  (constructed by Algorithm 1) and last term allocation list  $AL'$  as the input. It outputs the updating allocation list with the updating state  $upd$ . The *false* value of  $upd$  means there is no change for  $AL$ . The candidate list will be partitioned into two parts. Ones with no workers' proposal are the same as the last term ones, which is shown from line 7 to line 8. The others containing workers' proposals will be added to a new competing table with the winners of the last term. Each set of workers for applied tasks in competing table will be sorted by the utility value (compared by  $PCF(\hat{d}'_{x,a,b}, \hat{d}_{x,b}, \epsilon_{x,a}, \epsilon_{x,b})$ ) in descending order. The process is shown from line 10 to line 12.

By executing Algorithm 1 and Algorithm 2, we can construct our PUCE algorithm as shown in Algorithm 3. **The total task set and the total worker set can be divided into several time window slices. We execute execute PUCE on each time window in a batch-based style.** In the beginning, the not winning worker set  $NWW$  is initialized as the whole worker set  $W$ , and the allocation list  $AL$  is initialized as an empty set. We execute Algorithm 1 to get candidate allocation list  $CL$ . Then we execute Algorithm 2 to pick a new allocation list  $AL$  and get a updating state  $upd$ . When there are still

### Algorithm 3: PUCE

**Input:** Tasks  $\mathcal{T}$  and workers  $\mathcal{W}$  in the current time window  
**Output:** The task-worker matching pairs  $TWM$   
1 Initialize not winning worker set  $NWW$  as  $\mathcal{W}$ ;  
2 Initialize halt state  $hs$  as *false*;  
3 Initialize allocation list  $AL$  as  $m$  empty set list;  
4 **while**  $hs$  is not *true* **do**  
5     Get  $CL$  by executing Algorithm 1;  
6     Get  $AL$  and  $upd$  by executing Algorithm 2;  
7     Update  $NWW$  by removing new winners and adding new losers;  
8     Set  $hs = upd$ ;  
9 Set  $TWM$  as  $AL$ ;  
10 **return**  $TWM$ ;

TABLE III: Task-worker distances.

Worker/Task	$t_1$	$t_2$	$t_3$
$w_1$	12.2	3.61	17.12
$w_2$	5	10.44	12.21
$w_3$	9.43	18.25	7.28

some workers proposing to tasks ( $CL$  is not empty),  $upd$  will be set as *true*. We also update  $NWW$  by removing the new winner workers and adding the new loser workers. When no workers propose to any task,  $upd$  will be set as *false*. Thus we get the final task-worker matching pairs  $TWM$  as  $AL$ .

**Example 2** (Running Example of PUCE). *We give a running example of the whole process of PUCE following the motivation example. As shown in Figure 1, three workers  $w_1$ ,  $w_2$  and  $w_3$  have service areas 15, 15 and 10, respectively. Three tasks  $t_1$ ,  $t_2$  and  $t_3$  have task values 12.4, 11 and 13, respectively. The distance between each task and worker is shown in Table III.*

*Suppose there are three privacy budgets for each task-worker pair. The corresponding effective distance, the privacy budget and utility are shown in Table IV.*

*At the beginning,  $NWW$  is set as  $\{w_1, w_2, w_3\}$ .  $CL$  is set to NULL.  $w_1$  firstly judges whether the tasks within his service area will be added to the  $CL$ . He calculates the utility for  $t_1$  as  $U_1(1) = 0.1 > 0$  and adds  $\tilde{d}_{1,1}$  to  $CL[1]$ . Besides, he adds  $\tilde{d}_{2,1}$  to  $CL[2]$ . And  $w_2$ ,  $w_3$  also add their selected tasks (by the judgement in Algorithm 1). And we can get the data in  $CL$  as shown in Table V (the utility values are shown in square brackets).*

*Then we get  $CT$  by sorting  $CL$ , which is shown in Table VI.*

*After that, we find  $t_1$  is allocated to  $w_3$ . Besides,  $t_2$  and  $t_3$  fall into conflict for  $w_2$ . After the comparison of CEA,  $t_3$  is allocated to  $w_2$ . In the next round, there is only  $t_2$  unallocated. And  $w_1$  has not matched any task yet.  $w_1$  can only propose for  $t_2$ . However the utility of  $U_1(2)$  in this round is  $-3.1 \leq 0$ . Thus there is no worker proposing to any tasks in this round. And the process is end.*

**Privacy Analysis.** We define the query data set of worker  $w_j$  as  $X_j$ , which consists of all tasks in the service area of  $w_j$  (i.e.,  $R_j$ ). The neighboring data set of  $X_j$  is noted as  $X'_j$ . It satisfies that  $\|X_j - X'_j\| = 1$ , which means there is only one different task item between  $X_j$  and  $X'_j$ . We focus on the query  $f$  as ‘Get each distance from  $w_j$  to his service tasks  $R_j$ ’. That means  $f(X_j) = [d_{i_1,j}, \dots, d_{i_{|R_j|},j}]$ .

TABLE IV: Effective obfuscated distance, privacy budget and utility.

Matchable pair	$(d, \epsilon^{(1)})$ , utility	$(d, \epsilon^{(2)})$ , utility	$(d, \epsilon^{(3)})$ , utility
$(t_1, w_1)$	(12.7, 0.1)	0.1	(12.4, 0.3)
$(t_1, w_2)$	(5.5, 4.6)	2.8	(5.3, 4.65)
$(t_1, w_3)$	(9.93, 0.1)	2.87	(9.63, 0.4)
$(t_2, w_1)$	(4.11, 6.99)	0.4	(4.01, 7.1)
$(t_2, w_2)$	(10.94, 0.1)	0.46	(10.64, 0.2)
$(t_3, w_2)$	(12.71, 0.1)	0.69	(12.51, 0.3)
$(t_3, w_3)$	(7.78, 5.4)	0.32	(7.58, 5.5)

TABLE V: Candidate list  $CL$ .

CL			
1	$\tilde{d}_{1,1}=12.7, [0.1]$	$\tilde{d}_{1,2}=5.5, [2.8]$	$\tilde{d}_{1,3}=9.93 [2.87]$
2	$\tilde{d}_{2,1}=4.11, [0.4]$	$\tilde{d}_{2,2}=10.94, [0.46]$	
3	$\tilde{d}_{3,2}=12.71, [0.69]$	$\tilde{d}_{3,3}=7.78, [0.32]$	

TABLE VI: Competing table  $CT$ .

CT	1	2	3
1	$\tilde{d}_{1,3}=9.93 [2.87]$	$\tilde{d}_{1,2}=5.5, [2.8]$	$\tilde{d}_{1,1}=12.7, [0.1]$
2	$\tilde{d}_{2,2}=10.94, [0.46]$	$\tilde{d}_{2,1}=4.11, [0.4]$	
3	$\tilde{d}_{3,2}=12.71, [0.69]$	$\tilde{d}_{3,3}=7.78, [0.32]$	

**Theorem V.2.** *PUCE satisfies  $(\sum_{t_i \in R_j} b_{i,j} \epsilon_{i,j} r_j)$ -local differential privacy for each worker  $w_j$ .*

*Proof.* Let  $\mathcal{A}_j$  be the mechanism PUCE applying to  $w_j$  with query  $f$  defined above. Let  $X_j$  be the location of  $w_j$ . For query  $f(X_j) = [d_{i_1,j}, \dots, d_{i_{|R_j|},j}]$ , we extend it to an equivalent query  $\hat{f}(X_j) = f(X_j) \cdot \mathcal{J}$ , where  $\mathcal{J}$  is a block diagonal matrix:

$$\mathcal{J} = \begin{bmatrix} CP(b_{i_1,j}) & & & \\ & CP(b_{i_2,j}) & & \\ & & \ddots & \\ & & & CP(b_{i_{|R_j|},j}) \end{bmatrix}$$

Here,  $CP(b)$  is the compression of  $b$ , which means removing all zero element of  $b$ . For example, if  $b = [1, 1, 0, 0, 0]$ , then  $CP(b) = [1, 1]$ .  $\hat{f}(X_j)$  means query  $d_{i_u,j}$  for  $sum(b_{i_u,j})$  times for  $u \in [|R_j|]$ , where  $sum(b_{i_u,j})$  means the sum of all elements in  $b_{i_u,j}$ . We denote the size of  $\hat{f}(X_j)$  as  $|\hat{f}|$  and the  $a$ -th element of  $\hat{f}(X_j)$  as  $\hat{f}(X_j)_a$ .

Let  $Y_j$  denote the set of all published obfuscated distances of the worker  $w_j$  to tasks in  $R_j$ . Then we have  $Y_j = \hat{f}(X_j) + [\eta_1, \eta_2, \dots, \eta_{|\hat{f}|}]$ , where  $\eta_a (1 \leq a \leq |\hat{f}|)$  is an i.i.d random variable drawn from  $Lap(1/\epsilon_a)$ . Hence we have

$$\begin{aligned} \frac{\Pr[\mathcal{A}_j(X_j) = Y_j]}{\Pr[\mathcal{A}_j(X'_j) = Y_j]} &= \prod_{a \in [|\hat{f}|]} \left( \frac{\exp(-\epsilon_a |Y_{j,a} - \hat{f}(X_j)_a|)}{\exp(-\epsilon_a |Y_{j,a} - \hat{f}(X'_j)_a|)} \right) \\ &= \prod_{t_i \in R_j} \prod_{u \in [sum(b_{i,j})]} \left( \frac{\exp(-\epsilon_{i,j}^{(u)} |\tilde{d}_{i,j}^{(u)} - d_{i,j}|)}{\exp(-\epsilon_{i,j}^{(u)} |\tilde{d}_{i,j}^{(u)} - d'_{i,j}|)} \right) \\ &\leq \prod_{t_i \in R_j} \prod_{u \in [sum(b_{i,j})]} (\exp(\epsilon_{i,j}^{(u)} (|d_{i,j} - d'_{i,j}|))) \\ &= \prod_{t_i \in R_j} \exp(b_{i,j} \epsilon_{i,j} (|d_{i,j} - d'_{i,j}|)) \\ &\leq \exp\left(\sum_{t_i \in R_j} b_{i,j} \epsilon_{i,j} r_j\right). \end{aligned}$$

Because  $X_j$  contains only one element, then we have PUCE satisfies  $(\sum_{t_i \in R_j} b_{i,j} \epsilon_{i,j} r_j)$ -local differential privacy for each worker  $w_j$ .  $\square$

**Time Cost Analysis.** There are  $m$  task and  $n$  workers in  $Cond_{m,n}$ . Each of the  $n$  workers has  $Z$  privacy budget for each task. Therefore, the worst time cost for PUCE is  $O(m \cdot n \cdot Z)$ .

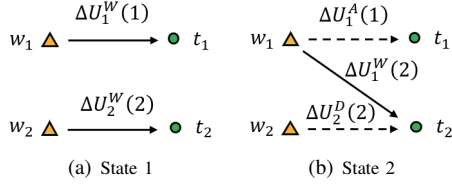


Fig. 2: Utility change.

## VI. PRIVATE GAME THEORETIC APPROACH (PGT)

In this section, we declare that each worker can compete for each task within their service area, whether or not they have already won a task. We modeled our problem as an exact potential game with at least one Nash equilibrium in pure strategy. To compare the utility values to make a choice, we approximate our utility function by replacing real distance with effective obfuscated distance.

### A. Cases of Utility Change in Competition

There are three cases of utility change in each time of competition for each task-worker pair. They are *Winning Change*, *Abandoned Change* and *Defeated Change*. We denote them as  $\Delta U_j^W(i)$ ,  $\Delta U_j^A(i)$  and  $\Delta U_j^D(i)$  respectively, which are expressed as follows:

$$\begin{aligned}\Delta U_j^W(i) &= v_i - f_d(\tilde{d}_{i,j}) - f_p(\epsilon_{i,j}^{(z)}), \\ \Delta U_j^A(i) &= -v_i + f_d(\tilde{d}_{i,j}), \\ \Delta U_j^D(i) &= -v_i + f_d(\tilde{d}_{i,j}).\end{aligned}$$

$\Delta U_j^W(i)$  means the utility change of winning task  $t_i$  for worker  $w_j$ .  $\Delta U_j^A(i)$  means the utility change of abandoning task  $t_i$  (because each worker can only match at most one task) for worker  $w_j$ .  $\Delta U_j^D(i)$  means the utility change of being defeated by some other competitor in competing for task  $t_i$  for worker  $w_j$ . It is the same with  $\Delta U_j^A(i)$ . We use  $\Delta U_j^{W(k)}(i)$ ,  $\Delta U_j^{A(k)}(i)$  and  $\Delta U_j^{D(k)}(i)$  to denote the above three utility change in  $k$ -th competition.

We give examples of these three utility changes. Suppose there are two workers  $w_1, w_2$  and two tasks  $t_1, t_2$ . At the first stage,  $w_1$  competes for  $t_1$  and  $w_2$  competes for  $t_2$ . Then the corresponding  $\Delta U_1^W(1)$  and  $\Delta U_2^W(2)$  are shown in Figure 2(a). At the second stage,  $w_1$  competes for  $t_2$  and gets it successfully. As is shown in Figure 2(b). The utility change between  $w_1$  and  $t_2$  is  $\Delta U_1^W(2)$ . The utility change between  $w_1$  and  $t_1$  is  $\Delta U_1^A(1)$ . The utility change between  $w_2$  and  $t_2$  is  $\Delta U_2^D(2)$ .

### B. Game Modeling and Nash Equilibrium

We approximate our PA-TA as *Privacy-aware Approximate Task Assignment* (PAA-TA) problem by replacing the real distance as effective distance. We formulate PAA-TA as an  $n$ -player strategic game,  $\mathcal{G} = \langle \mathcal{W}, \mathcal{S}, \mathcal{UT} \rangle$ .  $\mathcal{G}$  consists of players  $\mathcal{W}$ , strategy spaces  $\mathcal{S}$ , and utility functions  $\mathcal{UT}$ . We specify these three components as follows:

(1)  $\mathcal{W} = \{w_1, \dots, w_n\}$  denotes the finite set of  $n$  workers with  $n \geq 2$ . We will use worker and player interchangeably in the rest of the paper.

(2)  $\mathcal{S} = \{S_j\}_{j=1}^n$  is the strategy spaces (i.e., the overall strategy set of all players).  $S_j$  is the finite set of strategies

### Algorithm 4: PGT

---

**Input:** Tasks  $\mathcal{T}$  and workers  $\mathcal{W}$  in the current time window  
**Output:** The allocation list  $AL$

- 1 Initialize  $AL$  as a list with  $m$  null value;
- 2 Initialize halt state  $hs$  as *false*;
- 3 **while**  $hs$  is *false* **do**
- 4     Set  $hs$  as *true*;
- 5     **for each** worker  $w_j \in \mathcal{W}$  **do**
- 6         Get the maximal  $UT_j$  for each task  $t_i \in R_j \setminus \{AL[b]\}$ ;
- 7         **if**  $UT_j(st)$  is null or  $UT_j(st) \leq 0$  **then**
- 8             continue;
- 9         Set  $hs$  as *false*;
- 10         Set  $t_c$  as  $w_j$ 's already matched task;
- 11         Set  $t_b$  as the task with maximal  $UT_j$ ;
- 12         Set  $w_f$  as the worker matched  $t_b$  before;
- 13         Update effective distance-budget pair between  $t_b$  and  $w_j$ ;
- 14         Set  $AL[c] = \text{null}$ ;
- 15         Set  $AL[b] = w_j$ ;
- 16 **return**  $AL$ ;

---

available to worker  $w_j$ . Here, one strategy of worker  $w_j$  indicates an action that he proposes to some task  $t_i$  with a privacy budget  $\epsilon_{i,j}^{(u)}$  for the  $u$ -th proposal.

(3)  $\mathcal{UT} = \{UT_j^{(k)}\}_{j=1}^n$  is the utility functions of all players  $w_j$  where  $k$  is the total competition number. For each chosen strategy  $st \in \mathcal{S}$ ,  $UT_j^{(k)}(st) \in \mathbb{R}$  is the utility of player  $w_j$ . We calculate  $UT_j^{(k)}(st)$  as follows:

$$\begin{aligned}UT_j^{(k)}(st) &= \Delta U_j^{W(k)}(i_2) + \Delta U_{win(i_2)}^{D(k-1)}(i_2) + \Delta U_j^{A(k-1)}(i_1) \\ &= v_{i_2} - f_d(\tilde{d}_{i_2,j}^{(k)}) - f_p(\epsilon_{i_2,j}^{(z_k)}) - v_{i_2} + f_d(\tilde{d}_{i_2,win(i_2)}^{(k-1)}) \\ &\quad - v_{i_1} + f_d(\tilde{d}_{i_1,j}^{(k-1)}) \\ &= -f_d(\tilde{d}_{i_2,j}^{(k)}) - f_p(\epsilon_{i_2,j}^{(z_k)}) + f_d(\tilde{d}_{i_2,win(i_2)}^{(k-1)}) - v_{i_1} + f_d(\tilde{d}_{i_1,j}^{(k-1)}).\end{aligned}\tag{5}$$

In equation 5,  $w_j$  wins  $t_{i_1}$  and  $w_{win(i_2)}$  wins  $t_{i_2}$  in  $(k-1)$ -th competition.  $w_j$  will compete for  $t_{i_2}$  in  $k$ -th competition.

In the following part, we define *exact potential game* (EPG) and prove that PAA-TA is an EPG.

**Definition 7** (Exact Potential Game). A strategic game,  $\mathcal{G} = \langle \mathcal{W}, \mathcal{S}, \mathcal{UT} \rangle$ , is an Exact Potential Game (EPG) if there exists a function,  $\Phi : \mathcal{S} \rightarrow \mathbb{R}$ , such that for all  $st_j \in \mathcal{S}$ , it holds that,  $\forall w_j \in \mathcal{W}, \forall k \in N^+$ ,

$$\begin{aligned}UT_j^{(k)}(st'_j, st_{-j}) - UT_j^{(k)}(st_j, st_{-j}) \\ = \Phi^{(k)}(st'_j, st_{-j}) - \Phi^{(k)}(st_j, st_{-j}).\end{aligned}$$

**Theorem VI.1.** PAA-TA is an Exact Potential Game (EPG).

*Proof:* We define a potential function as

$$\Phi^{(k)}(st) = \sum_{t_i \in \mathcal{T}} \sum_{w_j \in \mathcal{W}} (s_{i,j}^{(k)} \cdot (v_i - f_d(\tilde{d}_{i,j})) - f_p(b_{i,j}^{(k)} \cdot \epsilon_{i,j}))$$

which represents the total utility value of the matching result in  $k$ -th competition that all worker gain. Let  $\tilde{U}_j^{(k)}(i) = v_i - f_d(\tilde{d}_{i,j}^{(k)}) - \sum_{t_i \in \mathcal{T}} \sum_{w_j \in \mathcal{W}} f_p(b_{i,j}^{(k)} \cdot \epsilon_{i,j})$  be the approximate value of  $U_j(i)$  by replacing the real distance  $d_{i,j}$  with the effective obfuscated distance  $\tilde{d}_{i,j}$ . Then we get the recurrence relation of  $\tilde{U}_j^{(k)}(i)$  for  $k$  as

$$\tilde{U}_j^{(k)}(i) = \begin{cases} \tilde{U}_j^{(k-1)}(i) + v_i - f_d(\tilde{d}_{i,j}^{(k)}) - f_p(\epsilon_{i,j}^{(z_k)}) & \#1 \\ \tilde{U}_j^{(k-1)}(i) - v_i + f_d(\tilde{d}_{i,j}^{(k-1)}) & \#2 \\ \tilde{U}_j^{(k-1)}(i) & \#3 \end{cases}$$

where condition #1 means  $w_j$  wins  $t_i$  in  $k$ -th competition, condition #2 means  $w_j$  gives up his original task or is defeated

in  $k$ -th competition and condition #3 means others. Suppose that  $w_j, w_{j_x}, w_{j_y}$  wins  $t_{i_1}, t_{i_2}, t_{i_3}$  in  $(k-1)$ -th competition respectively and  $w_j$  will compete for  $t_{i_2}$  ( $st_j$ ) or  $t_{i_3}$  ( $st'_j$ ) in  $k$ -th competition, then we obtain

$$\begin{aligned}
& \Phi^{(k)}(st'_j, st_{-j}) - \Phi^{(k)}(st_j, st_{-j}) \\
&= \tilde{U}_j^{(k)}(i_1) + \tilde{U}_j^{(k-1)}(i_2) + \tilde{U}_j^{(k)}(i_3) + \tilde{U}_{j_y}^{(k)}(i_3) + \tilde{U}_{j_x}^{(k-1)}(i_2) \\
&\quad - (\tilde{U}_j^{(k)}(i_1) + \tilde{U}_j^{(k)}(i_2) + \tilde{U}_j^{(k-1)}(i_3) + \tilde{U}_{j_y}^{(k-1)}(i_3) + \tilde{U}_{j_x}^{(k)}(i_2)) \\
&= \tilde{U}_j^{(k)}(i_3) - \tilde{U}_j^{(k-1)}(i_3) - (\tilde{U}_j^{(k)}(i_2) - \tilde{U}_j^{(k-1)}(i_2)) \\
&\quad + \tilde{U}_{j_y}^{(k)}(i_3) - \tilde{U}_{j_y}^{(k-1)}(i_3) - (\tilde{U}_{j_x}^{(k)}(i_2) - \tilde{U}_{j_x}^{(k-1)}(i_2)) \\
&= v_{i_3} - f_d(\tilde{d}_{i_3,j}^{(k)}) - f_p(\epsilon_{i_3,j}^{(z_k)}) - (v_{i_2} - f_d(\tilde{d}_{i_2,j}^{(k)}) - f_p(\epsilon_{i_2,j}^{(z_k)})) \\
&\quad - v_{i_3} + f_d(\tilde{d}_{i_3,j_y}^{(k-1)}) - (-v_{i_2} + f_d(\tilde{d}_{i_2,j_x}^{(k-1)})) \\
&= -f_d(\tilde{d}_{i_3,j}^{(k)}) - f_p(\epsilon_{i_3,j}^{(z_k)}) + f_d(\tilde{d}_{i_3,j_y}^{(k-1)}) \\
&\quad + f_d(\tilde{d}_{i_2,j}^{(k)}) + f_p(\epsilon_{i_2,j}^{(z_k)}) - f_d(\tilde{d}_{i_2,j_x}^{(k-1)}) \\
&= v_{i_3} - f_d(\tilde{d}_{i_3,j}^{(k)}) - f_p(\epsilon_{i_3,j}^{(z_k)}) - v_{i_3} + f_d(\tilde{d}_{i_3,j_y}^{(k-1)}) - v_{i_1} + f_d(\tilde{d}_{i_1,j}^{(k-1)}) \\
&\quad - (v_{i_2} - f_d(\tilde{d}_{i_2,j}^{(k)}) - f_p(\epsilon_{i_2,j}^{(z_k)})) - v_{i_2} + f_d(\tilde{d}_{i_2,j_x}^{(k-1)}) - v_{i_1} + f_d(\tilde{d}_{i_1,j}^{(k-1)}) \\
&= \Delta U_j^{W(k)}(i_3) + \Delta U_{j_y}^{D(k-1)}(i_3) + \Delta U_j^{A(k-1)}(i_1) \\
&\quad - (\Delta U_j^{W(k)}(i_2) + \Delta U_{j_x}^{D(k-1)}(i_2) + \Delta U_j^{A(k-1)}(i_1)) \\
&= UT_j^{(k)}(st'_j, st_{-j}) - UT_j^{(k)}(st_j, st_{-j})
\end{aligned}$$

According to Definition 7, the strategic game of the PAA-TA is an exact potential game. Therefore, PAA-TA has pure Nash equilibrium according to Theorem 2.3 in Ref [31].

**PGT Algorithm.** The server executes the competition process with the aid of workers. Each worker  $w_j$  needs to repeat choosing the best task  $t_b$  for the maximal utility value. If the maximal value is positive,  $w_j$  will update his effective distance-budget pair for  $t_b$  and ask the server to update the allocation list.

We give the process in Algorithm 4. The critical step is to calculate the best response information (maximal  $UT_j$ ) shown in line 6. The state variable  $hs$  is a boolean variable that indicates whether there still exists a task that can improve a utility function  $UT_j$  for any  $w_j \in \mathcal{W}$ . If there is no such task, the process will halt.

**Example 3** (Running Example of PGT). Consider the example in Table III, and the effective obfuscated distance and privacy budgets are shown in Table IV. As shown in Table VII, suppose in the  $k$ -th competition, the winners of  $t_1, t_2$  and  $t_3$  are  $w_1, w_2$  and  $w_3$  respectively. And  $w_1, w_2$  and  $w_3$  have consumed their first privacy budgets  $\epsilon_1$  for all three tasks. Besides, they public the obfuscated distances relevant to  $\epsilon_1$  for all tasks (so that all the effective obfuscated distances related to  $\epsilon_1$  are able to calculated by the server and all workers). Suppose  $f_d$  and  $f_p$  are both identity functions (i.e.,  $f_d(x) = x, f_p(x) = x$ ).

In the  $(k+1)$ -th competition, it is  $w_1$ 's turn to compete.  $w_1$  can only compete for  $t_2$ . He first uses his new privacy budget  $\epsilon_{2,1}^{(z_{k+1})} = \epsilon_{2,1}^{(2)} = 7.1$  and calculates the new effective obfuscated distance  $\tilde{d}_{2,1}^{(k+1)} = 4.01$ . After that, he calculates  $UT_1^{(k+1)} = -f_d(\tilde{d}_{2,1}^{(k+1)}) - f_p(\epsilon_{2,1}^{(z_{k+1})}) + f_d(\tilde{d}_{2,2}^{(k)}) - v_1 + f_d(\tilde{d}_{1,1}^{(k)}) = 0.13 > 0$ . Then, he publishes his privacy budget  $\epsilon_{2,1}^{(2)} = 7.1$  with the corresponding obfuscated distance  $\tilde{d}_{2,1}^{(2)}$  to the server. The server can also calculate the new effective obfuscated distance  $\tilde{d}_{2,1}^{(k+1)}$  and  $UT_1^{(k+1)}$ . It finds that  $UT_1^{(k+1)}$  is positive,

TABLE VII: Allocation list from the  $k$ -th competition.

Task	$k$ -th	$(k+1)$ -th	$(k+2)$ -th	$(k+6)$ -th
$t_1$	$w_1$	NULL		$w_2$
$t_2$	$w_2$	$w_1$		$w_1$
$t_3$	$w_3$	$w_3$		$w_3$

TABLE VIII: The timeline of effective distances and privacy budgets.

Pair/Times	$k$	$k+1$	$k+2$	$k+3$	$k+4$	$k+5$	$k+6$
$(t_1, w_1)$		(12.7,0.1)			(12.7,0.1) (12.4,0.3)		(12.7,0.1)
$(t_1, w_2)$		(5.5,4.6)	(5.5,4.6) (5.3,4.65)			(5.3,4.65)	
$(t_1, w_3)$		(9.93,0.1)		(9.93,0.1) (9.63,0.4)		(9.93,0.1)	(9.93,0.1) (9.63,0.4)
$(t_2, w_1)$	(4.11,6.99)	(4.11,6.99) (4.01,7.1)			(4.01,7.1)		
$(t_2, w_2)$			(10.94,0.1)			(10.94,0.1) (10.64,0.2)	(10.94,0.1)
$(t_3, w_2)$		(12.71,0.1)		(12.71,0.1)		(12.71,0.1) (12.51,0.3)	(12.71,0.1)
$(t_3, w_3)$				(7.78,5.4)			

which means  $w_1$  wins  $t_2$ . The server then alters the allocation table  $AL$  by setting the winner of  $t_2$  as  $w_1$  and the winner of  $t_1$  as NULL.

In the  $(k+2)$ -th competition, it is  $w_2$ 's turn to compete.  $w_2$  can compete for both  $t_1$  and  $t_3$ . He calculates  $UT_2^{(k+2)}[t_1] = v_1 - f_d(\tilde{d}_{1,2}^{(k+2)}) - f_p(\epsilon_{1,2}^{(z_{k+2})}) = 2.45 > 0$  and  $UT_2^{(k+2)}[t_3] = -f_d(\tilde{d}_{3,2}^{(k+2)}) - f_p(\epsilon_{3,2}^{(z_{k+2})}) + f_d(\tilde{d}_{3,3}^{(k+1)}) = -5.03 < 0$ . After that,  $w_2$  sets  $UT_2^{(k+2)}$  as  $UT_2^{(k+2)}[t_1]$ , which is the maximal positive value in set  $\{UT_2^{(k+2)}[t_1], UT_2^{(k+2)}[t_3]\}$ . Then,  $w_2$  applies to the server for  $t_1$  by proposing  $(\tilde{d}_{2,1}^{(2)}, \epsilon_{2,1}^{(2)})$ . After similar calculations, the server alters  $AL$  by setting the winner of  $t_1$  as  $w_2$ .

In the  $(k+3)$ -th competition, it is  $w_3$ 's turn to compete.  $w_3$  can only propose to  $t_1$ . However, the value  $UT_3^{(k+3)} = -9.95 < 0$ . Therefore,  $w_3$  does not compete for any tasks.

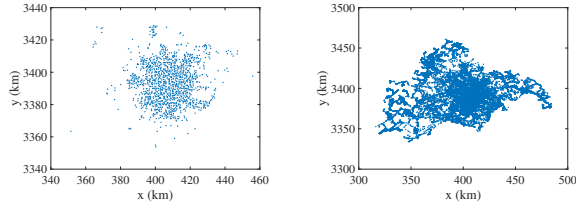
These three steps are repeated until all workers do not propose to any tasks (i.e., until the 6-th competition). Table VIII records the changing of effective obfuscated distances and privacy budgets. The red one (with  $UT > 0$ ) means there is a new winner who publishes a new privacy budget and updates the corresponding effective obfuscated distance. The green one (with  $UT \leq 0$ ) means the competitor fails to compete for the task and will publish neither his new obfuscated distance nor his new privacy budget.

**Convergence Analysis.** In order to answer the convergence speed of PGT, we need to know how many rounds it takes to find a pure Nash equilibrium. For the corresponding potential game of a PAA-TA instance,  $\mathcal{G} = \langle \mathcal{W}, \mathcal{S}, UT \rangle$ , we assume there is an equivalent game with potential function  $\Phi_{\mathbb{Z}}(st) = d \cdot \Phi(st)$ , where  $d$  is a positive multiplicative factor satisfying that  $\Phi_{\mathbb{Z}}(st) \in \mathbb{Z}$  for  $\forall st \in \mathcal{S}$ . Let  $st^*$  be the best strategy the workers can choose in this PAA-TA game instance. Based on the above assumption, we prove that PGT executes at most  $\Phi_{\mathbb{Z}}(st^*)$  rounds.

**Theorem VI.2.** PGT executes at most  $\Phi_{\mathbb{Z}}(st^*)$  rounds to achieve a pure Nash equilibrium, where  $\Phi_{\mathbb{Z}}(st^*) = d \cdot \Phi(st^*)$  is a scaled potential function with integer value  $d$  and  $st^*$  is the optimal strategy the workers can choose in the potential PAA-TA game instance.

*Proof.* We say PGT converges when no workers deviate from their current strategies. If PGT has not converged, then at





(a) Order location distribution (b) Taxi location distribution

Fig. 3: Orders and taxis of Chengdu Didi data set.

least one worker  $w_j$  deviates from his current strategy in each round. Besides the new change strategy  $st'_j$  of  $w_j$  is better than his current strategy  $st_j$ . And the change will improve at least 1 (i.e.,  $\Phi_{\mathbb{Z}}(st'_j, s_{-j}) - \Phi_{\mathbb{Z}}(st_j, s_{-j}) \geq 1$ ) for potential games. Because the maximum value of scaled potential function is  $\Phi_{\mathbb{Z}}(st^*)$ , and the total utility is always positive, PGT needs at most  $\Phi_{\mathbb{Z}}(st^*)$  rounds to converge to a pure Nash equilibrium.  $\square$

**Quality Analysis.** Since the distance in our game is rather real distance than effective obfuscated distance, here we give the upper bound of expectation of *price of stability* (EPoS) and the lower bound of expectation of *price of anarchy* (EPoA). Let

$$\begin{aligned}
 U_j^L(i) &= v_i - f_d(d_{i,j}) - f_p(\sum_{t_k \in R_j} \text{sum}(\epsilon_{k,j})) \\
 U_j^H(i) &= v_i - f_d(d_{i,j}) - f_p(\min(\epsilon_{i,j})) \\
 U_{min}^+(i) &= \begin{cases} \min_{R_j \ni t_i, U_j^L(i) > 0} U_j^L(i), & \text{if there exists } U_j^L(i) > 0 \\ 0, & \text{otherwise} \end{cases} \\
 U_{max}^+(i) &= \begin{cases} \max_{R_j \ni t_i} U_j^H(i), & \text{if there exists } U_j^H(i) > 0 \\ 0, & \text{otherwise} \end{cases}
 \end{aligned}$$

Then we have Theorem VI.3 as follows.

**Theorem VI.3.** *In the strategic game of PGT, the lower bound of EPoA is  $\frac{\sum_{t_i \in T} U_{min}^+(i)}{\sum_{t_i \in T} U_{max}^+(i)}$  ( $\sum_{t_i \in T} U_{max}^+(i) \neq 0$ ) and the upper bound of EPoS is 1.*

Please refer to details of the proof of Theorem VI.3 in Appendix B in our technical report [30].

**Theorem VI.4.** *PGT satisfies  $(\sum_{t_i \in R_j} b_{i,j} \epsilon_{i,j} r_j)$ -local differential privacy for each worker  $w_j$ .*

The proof is similar to Theorem V.2, and please refer to the details in Appendix C in our technical report [30].

## VII. EXPERIMENT

### A. Data Sets

We test our mechanisms in real and synthetic data sets.

**Real Data Set.** We use Didi Chuxing[32] in Chengdu, China, as our real data set. We choose the day with the most requests for evaluation (November 18, 2016 in Chengdu) and perform the same preprocess in the existing work [33], which is denoted as *chengdu*.

*Chengdu* contains 259347 orders and 30000 taxis. Each order tuple is a taxi request consisting of a release time, a pickup location, a drop-off location, and some passengers. Each taxi tuple is a basic message consisting of the original location of the taxi and its capacity. The location distribution of taxis is shown in Figure 3(b).

TABLE IX: Methods.

	Private version	Non-Private version	Non-PPCF version
Distance Elimination	PDCE [3]	DCE	PDCE-nppcf
Utility Elimination	PUCE	UCE	PUCE-nppcf
Game Theory	PGT	GT	—
Greedy	—	GRD	—

TABLE X: Experimental settings.

Parameters	Values
worker-task ratio	1, 1.5, 2, 2.5, 3
task values	1.5, 3, 4.5, 6, 7.5
worker range	0.8, 1.1, 1.4, 1.7, 2.0
privacy budget	[0.5, 0.75], [0.75, 1.00], [1.00, 1.25], [1.25, 1.50], [1.50, 1.75]; <b>[0.5, 1.75]</b>
privacy budget group size	<b>7</b>

**Synthetic Data Set.** We generate two data sets with 2-dimensional uniform distribution and normal distribution, respectively. For the uniform distribution data set, we randomly generate 300000 points for tasks and 900000 for workers in a plane with a range of  $100 \times 100$ . Each point follows a 2-dimensional uniform distribution with an average of 0.

For the normal distribution data set, we generate 300k and 900k points for tasks and workers, respectively. The expectation and variance for all points are 0 and 150, respectively.

### B. Experimental Setup

We split the orders into batches by timestamp. Each batch contains at most 1000 orders. Figure 3(a) is a batch example of the order distribution. We also split the taxis into ten groups for the real data set, each containing 3000 taxis. We use each worker group circularly for each batch. We set the start locations of orders as task locations and the start locations of taxis as worker locations.

Let  $S_T$  and  $S_W$  be two sets for tasks and workers. We define the value  $p_{wt} = \frac{|S_W|}{|S_T|}$  as *worker-task ratio*, which stands for the ratio between worker number, and task number.

We alter the method in Ref [3] by constraining the workers' proposing range and replacing the PCF with PPCF in order to get reasonable comparison with our PUCE. We denote this altering method in Ref [3] as Private Distance Conflict-Elimination (PDCE). The different between PUCE and PDCE is the optimization objective. In PDCE, the goal is to minimize all the travelling distance on the platform, which only consider the distance variable. However in PUCE, the goal is to maximize the utility function of the platform, which considers the task value, travel distance and privacy budget.

We compare our PUCE and PGT with PDCE. Besides, we construct the non-privacy solution of each privacy solution by eliminating the privacy budget cost in the utility function and replacing obfuscated distance with real distance. These non-privacy solutions are Utility Conflict-Elimination (UCE), Game Theory (GT), Distance Conflict-Elimination (DCE) and Greedy (GRD). Here, GRD always greedily chooses the current best worker-task pair (with the highest utility) for each worker. We also construct the non-PPCF solution of PUCE and PDCE by replacing the PPCF part with the PCF part. We denote these non-PPCF solutions as PUCE-nppcf and PDCE-nppcf. We compare all these methods above and summarize them in Table IX.

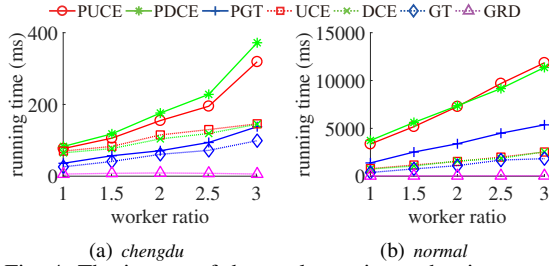


Fig. 4: The impact of the worker ratio on the time cost.

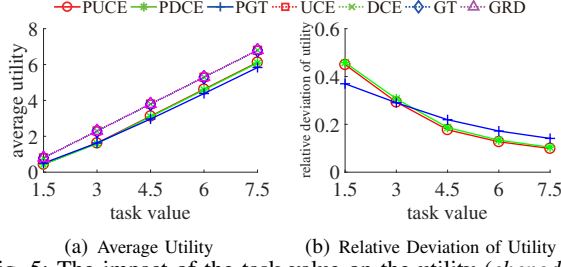


Fig. 5: The impact of the task value on the utility (*chengdu*).

We show the parameter settings in table X, where the default values are marked in bold.

As for distance value function  $f_d$  and privacy budget value function  $f_p$ , we model them as linear functions and use  $f_d(x) = \alpha x$  and  $f_p(x) = \beta x$  in our experiment. We set  $\alpha = 1$  and  $\beta = 1$ .

We run our experiment on an Intel(R) Xeon(R) Silver 4210R CPU @ 2.4GHz with 128 GB RAM in Java.

### C. Measures

We design a utility-based empirical measure of the efficiency of our proposed mechanisms.

**Average Utility:** We define the average utility  $U_{AVG}$  as  $\frac{\sum_{(i,j) \in M} U_j(i)}{|M|}$ , which means the average utility value of a successful task-worker pair.

**Relative Deviation of Utility:** Let the utility of non-privacy solutions be  $U_{NP}$  and privacy ones be  $U_P$ . We define the relative deviation of utility  $U_{RD}$  as  $\frac{U_{NP} - U_P}{U_{NP}}$ .

**Average Travel Distance:** We define the average travel distance  $D_{AVG}$  as  $\frac{\sum_{(i,j) \in M} d_{i,j}}{|M|}$ , which means the average travel distance of a successful task-worker pair.

**Relative Deviation of Distance:** Let the distance of non-privacy solutions be  $D_{NP}$  and privacy ones be  $D_P$ . We define the relative deviation of distance  $D_{RD}$  as  $\frac{D_P - D_{NP}}{D_{NP}}$ .

### D. Experimental Result

1) *Time Cost*: Figure 4 shows the time cost on different worker ratio from 1 to 3 while the other parameters are in the default values in Table X. We can see that the time cost increases linearly with the worker ratio. That is because when we fix the task quantity, as the worker ratio becomes larger, the competition between workers will become more fierce, and it will cost more time to finish the whole competition.

Besides, we can find that PUCE costs nearly the same time over the change of worker ratio. PGT costs much less time than PUCE and PDCE. Compared with PDCE, PGT costs about 52%–63% less time in *chengdu* and 50%–63% in *normal*.

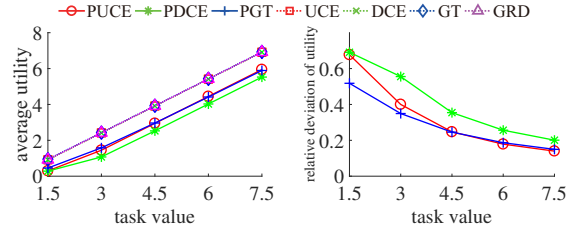


Fig. 6: The impact of the task value on the utility (*normal*).

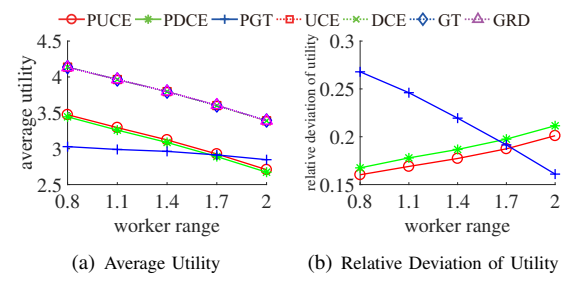


Fig. 7: The impact of the worker range on the utility (*chengdu*).

2) *Average Utility*: Figure 5 and 6 show the relation between the utility and the task value on *chengdu* and *normal* respectively. We change the task value from 1.5 to 7.5 and set other parameters as the default values.

In Figure 5(a) and 6(a), the utility increases approximately linear with the task value. We can see that PGT performs worse than PDCE slightly in *chengdu*, but better in *normal*. PGT even performs better than PUCE in *normal*. The reason is that PGT takes advantage over the other two when the workers' service area contains many tasks. The data in *chengdu* is of road network data which is relative sparse than that in *normal*. Thus fixed the service area, a worker in *chengdu* can propose to less tasks than that in *normal* on average, which leads to poor utility for PGT. (The following experiment result in Figure 7(a) proves this inference.) PUCE performs better than PDCE in both of the two data sets. The relative deviation of utility impacted by the task value is shown in Figure 5(b) and 6(b). We can see that the relative deviation of utility decreases with the task value increase from 1.5 to 7.5, which means the absolute deviation between the private and non-private solutions keeps nearly stable. And when the task value becomes larger and larger, the utility of private solutions equals that of non-private solutions asymptotically.

Figure 7 and 8 show the relation between the utility and the worker range on *chengdu* and *normal* respectively. The worker service area (denoted as worker range) increases from 0.8 to 2, and the other parameters are set as default values. The average utility depends on the total utility and the matching quantity. Specifically, in Figure 7(a), the average utility of all solutions decreases when the worker range increases from 0.8 to 2. It is because when worker service areas become larger, more workers (who have no task to propose to in some small range conditions, denoting them as  $\mathcal{W}_L$ ) will be able to propose to some tasks. With the ratio of  $\mathcal{W}_L$  becoming larger, the average distance to all matching tasks becomes larger, making the average utility smaller.

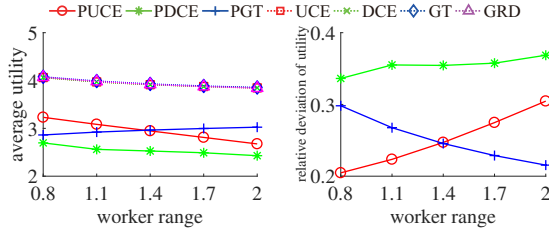


Fig. 8: The impact of the worker range on the utility (*normal*).

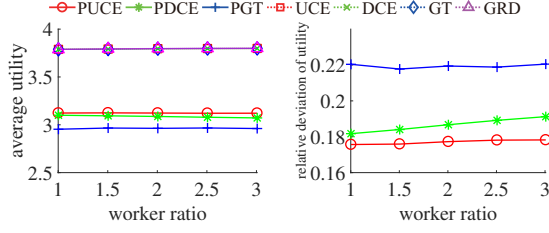


Fig. 9: The impact of the worker ratio on the utility (*chengdu*).

Besides, we can see that the utility of PGT decreases slower than both PUCE and PDCE. The utility of PGT is no less 88% when the worker range is no more than 1.6. And as the worker range increases, the utility of PGT will exceed the other two. The reason why PGT keeps lower decrease is that PGT can avoid ineffective competition. When the service area becomes larger, the competition becomes more intense, and the advantage of PGT becomes more apparent.

Figure 7(b) shows the relative deviation of utility affected by the worker range. We can see that the utility of PGT will tend to that of its non-privacy solution when the worker range becomes larger and larger. However PUCE and PDCE deviate more as the worker range becomes larger. That is because when the worker's service area becomes larger, it has a greater possibility of disturbing a large real distance to a small obfuscated distance or a small real distance to a large obfuscated distance. Without the guarantee of total utility function  $ST$ , the total proposing workers' utilities in PUCE and PDCE both decrease dramatically when the worker range increases.

From Figure 8(a), we can get the similar conclusion to that in Figure 7(a). Besides, we can find when the worker range becomes large enough, the decline rate of average utility for PUCE and PDCE tend to be small. That is because being too far away will make the utility value non-positive, and the server will not choose. The average utility of PGT increases slightly, which is 16% larger than PDCE on average. That is because PGT can increase the total utility more rapidly than the matching quantity.

Figure 9 and 10 show the relation between the utility and the worker ratio. We set the worker range and task value as default values in Table X. From figure 9(a) and 10(a), we can see that the worker ratio does not affect the average utility very much. That is because the increase of workers does not significantly increase proposing workers. Besides, we can see that PUCE always keeps a higher average utility than PDCE.

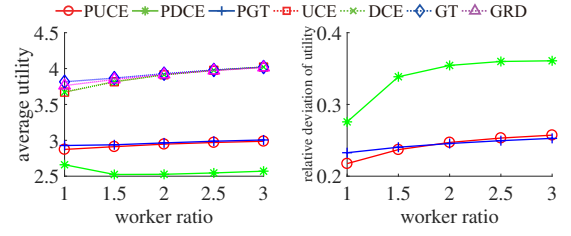


Fig. 10: The impact of the worker ratio on the utility (*normal*).

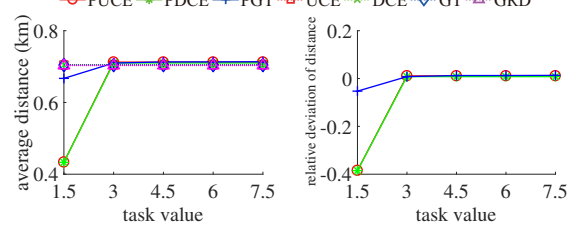


Fig. 11: The impact of the task value on the distance (*chengdu*).

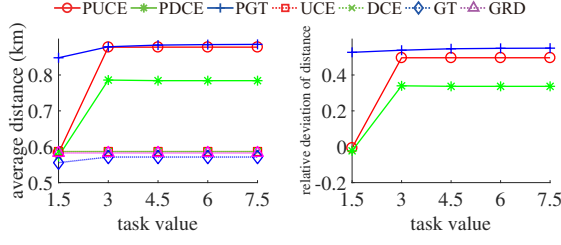
And PGT performs worse than PDCE in *chengdu* but better in *normal*.

3) *Average Travel Distance*: Figure 11 to Figure 16 show the influence of the task value, worker range and worker ratio on the distance. PDCE is better than PUCE and PGT in most cases. That is because the goal of PDCE is only to minimize the total travel distance on the platform without considering task value and privacy budget cost. Besides, we can see that different data sets lead to different comparison results for PUCE, PGT and PDCE. The average travel distance of PDCE on *normal* outperforms the other two on *chengdu*.

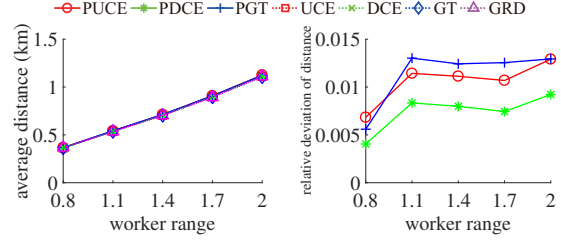
Figure 11 and 12 show the relation between the average distance and the task values. We can see that task values do not affect the average distance when the task value is larger than 3. That is because when the task value is large enough, it will not affect the difference between the two utility values. Workers will not choose many tasks in their range when the task value is minimal, leading to a small average distance. Besides, PUCE is better than PGT slightly but worse than PDCE. However the difference of the distances between PUCE and PDCE keeps stable as task value increases.

Figure 13 and 14 show the relation between the average distance and the worker service area. We can see that the average distance increases when worker range increases. That is because a larger range will lead to more proposing workers with far distance, making the average distance larger. The average distances of PUCE and PGT are nearly equal. They are worse than PDCE with nearly fix difference distance value when the worker range is larger than 1.4.

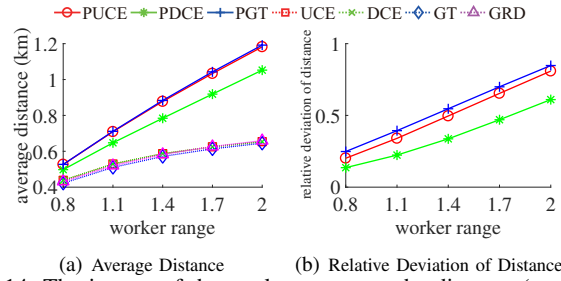
Figure 15 and 16 show the relation between the average distance and the worker ratio. Especially in figure 15(a), the average distance in non-privacy solutions decreases when the worker ratio increases. That is because, with the increase in workers, the competition has become rigorous. The number of tasks limits the increase of workers' proposals, and a task will be allocated to the worker at a small distance. Therefore,



(a) Average Distance (b) Relative Deviation of Distance  
Fig. 12: The impact of the task value on the distance (*normal*).



(a) Average Distance (b) Relative Deviation of Distance  
Fig. 13: The impact of the worker range on the distance (*chengdu*).

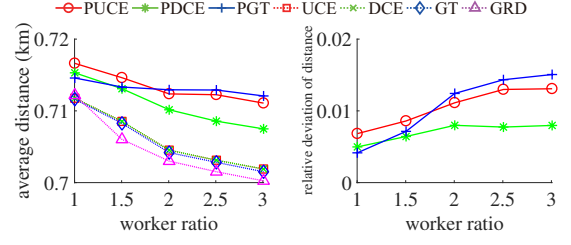


(a) Average Distance (b) Relative Deviation of Distance  
Fig. 14: The impact of the worker range on the distance (*normal*). the average becomes smaller with the worker ratio becoming larger. As for privacy solutions, competition will also cost more privacy budget on utility value, which will relieve the reduction in privacy solutions. **Similar to the comparison result before, PDCE is better than the other two schemes when the worker ratio is larger than 1.5.**

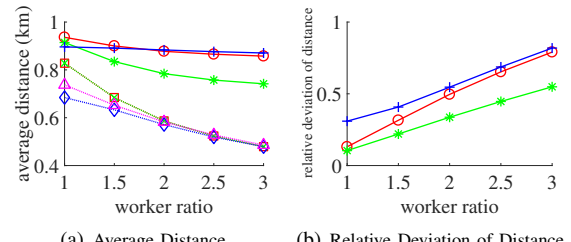
4) *PPCF and Non-PPCF*: We compare our PUCE and the PDCE with non-PPCF ones (PUCE-nppcf and PDCE-nppcf). We fix the task value as 4.5, the worker range distance as 1.4, and the worker ratio as 2. We divide the privacy budget range into 5 groups shown in Table X.

Figure 17 shows the relation between the average utility and the privacy budget. We mark the median of each interval as the value of the x-axis.

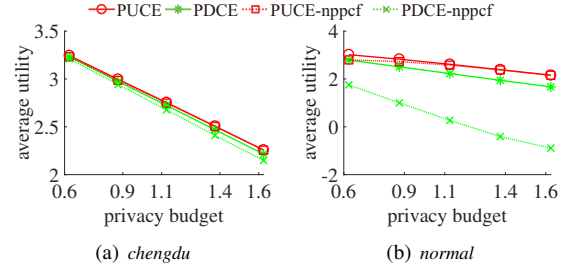
The solutions with PPCF are better than that without PPCF when the privacy budget is small. It means PPCF is suitable for high-privacy situations and is continuously more effective than that without PPCF. **As the privacy budget increases, the average utility decreases. That is because large privacy budgets give large average privacy budget cost for workers. Although high privacy budgets are able to lead high utility match, it also leads to high privacy budget cost.** Besides, as the privacy budget increases, the difference between PPCF and non-PPCF is eliminated. That is because the larger the privacy budget, the more accurate the obfuscated distance, and the smaller difference between PPCF and PCF.



(a) Average Distance (b) Relative Deviation of Distance  
Fig. 15: The impact of the worker ratio on the distance (*chengdu*).



(a) Average Distance (b) Relative Deviation of Distance  
Fig. 16: The impact of the worker ratio on the distance (*normal*).



(a) *chengdu* (b) *normal*  
Fig. 17: The impact of privacy on the utility.

## VIII. CONCLUSION

In this paper, we formalize the Privacy-aware Task Assignment (PA-TA) Problem, which assigns a task to a worker to get a high utility value. In order to make use of obfuscated distance published by workers, we propose new notations called *effective obfuscated distance* and *effective privacy budget*. To get a higher utility value, we offer a new comparison function called PPCF and prove that it can achieve better effectiveness than PCF in both theory and practice. Besides, we propose another game theoretic approach to solve the problem. Extensive experiments have been conducted to show the efficiency and effectiveness of our methods on both real and synthetic data sets.

Our PUCE and PGT only consider the distance privacy of one worker in his service area. If the service area of a worker is small enough and the quantity of tasks in this area is large enough, attackers can locate the worker's position through trilateration by viewing the entire area as a position. That is because too much effective obfuscated distance from a worker to many tasks will outline the worker's service area. Our subsequent work will focus on this problem and consider how to hide correlation privacy caused by the relation between different worker service areas. **Besides, in our goal function, we suppose the task value is only related to the task itself. Our subsequent work will extract the payment from the task value and research on the suppose that: the task value is related to task itself, travel distance and privacy cost.**



## REFERENCES

- [1] C. Dwork, "Differential privacy," in *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II* (M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, eds.), vol. 4052 of *Lecture Notes in Computer Science*, pp. 1–12, Springer, 2006.
- [2] H. To, C. Shahabi, and L. Xiong, "Privacy-preserving online task assignment in spatial crowdsourcing with untrusted server," in *34th IEEE International Conference on Data Engineering, ICDE 2018, Paris, France, April 16-19, 2018*, pp. 833–844, IEEE Computer Society, 2018.
- [3] 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.
- [4] M. Li, J. Wang, L. Zheng, H. Wu, P. Cheng, L. Chen, and X. Lin, "Privacy-preserving batch-based task assignment in spatial crowdsourcing with untrusted server," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 947–956, 2021.
- [5] D. Deng, C. Shahabi, and U. Demiryurek, "Maximizing the number of worker's self-selected tasks in spatial crowdsourcing," in *21st SIGSPATIAL International Conference on Advances in Geographic Information Systems, SIGSPATIAL 2013, Orlando, FL, USA, November 5-8, 2013* (C. A. Knoblock, M. Schneider, P. Kröger, J. Krumm, and P. Widmayer, eds.), pp. 314–323, ACM, 2013.
- [6] L. Zhang, T. Hu, Y. Min, G. Wu, J. Zhang, P. Feng, P. Gong, and J. Ye, "A taxi order dispatch model based on combinatorial optimization," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, pp. 2151–2159, ACM, 2017.
- [7] 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 *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pp. 2245–2252, AAAI Press, 2019.
- [8] Y. Tong, J. She, B. Ding, L. Wang, and L. Chen, "Online mobile micro-task allocation in spatial crowdsourcing," in *32nd IEEE International Conference on Data Engineering, ICDE 2016, Helsinki, Finland, May 16-20, 2016*, pp. 49–60, IEEE Computer Society, 2016.
- [9] Y. Wang, Y. Tong, C. Long, P. Xu, K. Xu, and W. Lv, "Adaptive dynamic bipartite graph matching: A reinforcement learning approach," in *35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8-11, 2019*, pp. 1478–1489, IEEE, 2019.
- [10] L. Kazemi and C. Shahabi, "Geocrowd: enabling query answering with spatial crowdsourcing," in *SIGSPATIAL 2012 International Conference on Advances in Geographic Information Systems (formerly known as GIS), SIGSPATIAL'12, Redondo Beach, CA, USA, November 7-9, 2012* (I. F. Cruz, C. A. Knoblock, P. Kröger, E. Tanin, and P. Widmayer, eds.), pp. 189–198, ACM, 2012.
- [11] H. To, C. Shahabi, and L. Kazemi, "A server-assigned spatial crowdsourcing framework," *ACM Trans. Spatial Algorithms Syst.*, vol. 1, no. 1, pp. 2:1–2:28, 2015.
- [12] J. She, Y. Tong, L. Chen, and C. C. Cao, "Conflict-aware event-participant arrangement and its variant for online setting," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 9, pp. 2281–2295, 2016.
- [13] P. Cheng, X. Lian, L. Chen, J. Han, and J. Zhao, "Task assignment on multi-skill oriented spatial crowdsourcing," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 8, pp. 2201–2215, 2016.
- [14] R. M. Karp, U. V. Vazirani, and V. V. Vazirani, "An optimal algorithm for on-line bipartite matching," in *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA* (H. Ortiz, ed.), pp. 352–358, ACM, 1990.
- [15] W. Ni, P. Cheng, L. Chen, and X. Lin, "Task allocation in dependency-aware spatial crowdsourcing," in *36th IEEE International Conference on Data Engineering, ICDE 2020, Dallas, TX, USA, April 20-24, 2020*, pp. 985–996, IEEE, 2020.
- [16] Y. Zhao, K. Zheng, J. Guo, B. Yang, T. B. Pedersen, and C. S. Jensen, "Fairness-aware task assignment in spatial crowdsourcing: Game-theoretic approaches," in *37th IEEE International Conference on Data Engineering, ICDE 2021, Chania, Greece, April 19-22, 2021*, pp. 265–276, IEEE, 2021.
- [17] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3-4, pp. 211–407, 2014.
- [18] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: differential privacy for location-based systems," in *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013* (A. Sadeghi, V. D. Gligor, and M. Yung, eds.), pp. 901–914, ACM, 2013.
- [19] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pp. 429–438, IEEE Computer Society, 2013.
- [20] 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.
- [21] J. S. Kim, Y. D. Chung, and J. W. Kim, "Differentially private and skew-aware spatial decompositions for mobile crowdsensing," *Sensors*, vol. 18, no. 11, p. 3696, 2018.
- [22] L. Wang, D. Zhang, D. Yang, B. Y. Lim, and X. Ma, "Differential location privacy for sparse mobile crowdsensing," in *IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain* (F. Bonchi, J. Domingo-Ferrer, R. Baeza-Yates, Z. Zhou, and X. Wu, eds.), pp. 1257–1262, IEEE Computer Society, 2016.
- [23] N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Optimal geo-indistinguishable mechanisms for location privacy," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014* (G. Ahn, M. Yung, and N. Li, eds.), pp. 251–262, ACM, 2014.
- [24] X. Jin and Y. Zhang, "Privacy-preserving crowdsourced spectrum sensing," *IEEE/ACM Trans. Netw.*, vol. 26, no. 3, pp. 1236–1249, 2018.
- [25] A. Ghosh and A. Roth, "Selling privacy at auction," in *Proceedings 12th ACM Conference on Electronic Commerce (EC-2011), San Jose, CA, USA, June 5-9, 2011* (Y. Shoham, Y. Chen, and T. Roughgarden, eds.), pp. 199–208, ACM, 2011.
- [26] K. Nissim, C. Orlandi, and R. Smorodinsky, "Privacy-aware mechanism design," in *Proceedings of the 13th ACM Conference on Electronic Commerce, EC 2012, Valencia, Spain, June 4-8, 2012* (B. Faltings, K. Leyton-Brown, and P. Ipeirotis, eds.), pp. 774–789, ACM, 2012.
- [27] D. Xiao, "Is privacy compatible with truthfulness?," in *Innovations in Theoretical Computer Science, ITCS '13, Berkeley, CA, USA, January 9-12, 2013* (R. D. Kleinberg, ed.), pp. 67–86, ACM, 2013.
- [28] R. E. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems*. SIAM, 2009.
- [29] I. J. Myung, "Tutorial on maximum likelihood estimation," *Journal of mathematical Psychology*, vol. 47, no. 1, pp. 90–100, 2003.
- [30] "Dynamic private task assignment under differential privacy [technical report]," <https://cspcheng.github.io/pdf/GameDP-SC.pdf>.
- [31] Y. H. Chew, B.-H. Soong, et al., *Potential game theory*. Springer, 2016.
- [32] Didi Chuxing. <http://www.didichuxing.com/>.
- [33] Y. Tong, Y. Zeng, Z. Zhou, L. Chen, J. Ye, and K. Xu, "A unified approach to route planning for shared mobility," *Proc. VLDB Endow.*, vol. 11, no. 11, pp. 1633–1646, 2018.
- [34] S. P. Liew, T. Takahashi, S. Takagi, F. Kato, Y. Cao, and M. Yoshikawa, "Network shuffling: Privacy amplification via random walks," in *SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022* (Z. Ives, A. Bonifati, and A. E. Abbadi, eds.), pp. 773–787, ACM, 2022.

## IX. APPENDIX

### A. Proof for Theorem V.1

Before the proof of Theorem V.1, we declare and prove Lemma IX.1 and Lemma IX.2 as follows.

**Lemma IX.1.** *For any  $d_x, d_y, \epsilon_x, \epsilon_y$ ,  $\hat{d}_x = d_x + \text{Lap}(0, 1/\epsilon_x)$ ,  $\hat{d}_y = d_y + \text{Lap}(0, 1/\epsilon_y)$ , we have  $\text{PCF}(\hat{d}_x, \hat{d}_y, \epsilon_x, \epsilon_y) > \frac{1}{2} \Leftrightarrow \hat{d}_x < \hat{d}_y$ .*

*Proof.* Let  $\eta_x \sim \epsilon_x, \eta_y \sim \epsilon_y$ . Then we have

$$\text{PCF}(\hat{d}_x, \hat{d}_y, \epsilon_x, \epsilon_y) = \iint_{D_R} f(\eta_x, \eta_y)$$

where  $f(\eta_x, \eta_y) = \frac{\epsilon_x \epsilon_y}{4} e^{-\epsilon_x |\eta_x| - \epsilon_y |\eta_y|}$  and  $D_R$  is the plane set satisfying  $D_R = \{(\eta_x, \eta_y) : \eta_y - \eta_x < \hat{d}_y - \hat{d}_x\}$ . Note that  $f(\eta_x, \eta_y)$  is symmetry about both x-axis and y-axis and  $D_R$  is part of plane split by line  $l_\eta : \eta_y = \eta_x + \hat{d}_y - \hat{d}_x$ . So we know that only when  $l_\eta$  crosses the origin ( $\hat{d}_y = \hat{d}_x$ ),  $PCF(\hat{d}_x, \hat{d}_y, \epsilon_x, \epsilon_y)$  equals  $\frac{1}{2}$ . When  $\hat{d}_y - \hat{d}_x > 0$ ,  $PCF(\hat{d}_x, \hat{d}_y, \epsilon_x, \epsilon_y) < \frac{1}{2}$ , and  $\hat{d}_y - \hat{d}_x < 0$ ,  $PCF(\hat{d}_x, \hat{d}_y, \epsilon_x, \epsilon_y) > \frac{1}{2}$ . Therefore,  $PCF(\hat{d}_x, \hat{d}_y, \epsilon_x, \epsilon_y) > \frac{1}{2} \Leftrightarrow \hat{d}_x < \hat{d}_y$ .  $\square$

**Lemma IX.2.** For any two continue and differentiable non-negative functions  $f, g$  defined in  $\mathbb{R}$ , if there exists an interval  $[a, +\infty)$  satisfying that  $\int_a^{+\infty} f(x)dx = \int_a^{+\infty} g(x)dx$  and there exists a point  $x_0 \in (a, +\infty)$  satisfying  $f(x) \geq g(x)$  for  $x \in (a, x_0]$  and  $f(x) \leq g(x)$  for  $x \in (x_0, +\infty)$ , then  $\int_a^x f(x)dx \geq \int_a^x g(x)dx$  for all  $x \in [a, +\infty)$ .

*Proof.* For any  $x \in [a, +\infty)$ , we can divide it into two cases: (1)  $x \in [a, x_0]$ ; (2)  $x \in (x_0, +\infty]$ . If (1) holds, according to  $f(x) \geq g(x)$  for  $x \in (a, x_0]$ , we have directly get

$$\int_a^x f(x)dx \geq \int_a^x g(x)dx \quad \text{for } x \in [a, x_0]. \quad (6)$$

If (2) holds, then we have  $\int_{x_0}^{+\infty} f(x)dx \leq \int_{x_0}^{+\infty} g(x)dx$ . And we can get

$$\begin{aligned} & \int_a^x f(x)dx - \int_a^x g(x)dx \\ &= \int_a^{+\infty} f(x)dx - \int_{x_0}^{+\infty} f(x)dx - \left( \int_a^{+\infty} g(x)dx - \int_{x_0}^{+\infty} g(x)dx \right) \\ &= \int_{x_0}^{+\infty} g(x)dx - \int_{x_0}^{+\infty} f(x)dx \geq 0. \end{aligned}$$

So we have

$$\int_a^x f(x)dx \geq \int_a^x g(x)dx \quad \text{for } x \in (x_0, +\infty). \quad (7)$$

From Equation 6 and 7, we can have  $\int_a^x f(x)dx \geq \int_a^x g(x)dx$  for  $x \in [a, +\infty)$ .  $\square$

Based on Lemma IX.1 and Lemma IX.2, we give the proof of Theorem V.1 as follows.

*Proof.* From Lemma IX.1, we have  $PCF(\hat{d}_x, \hat{d}_y, \epsilon_x, \epsilon_y) > \frac{1}{2} \Leftrightarrow \hat{d}_x < \hat{d}_y$ . From Equation 3, we have  $\Pr[d_x < d_y] > \frac{1}{2} \Leftrightarrow d_x < \hat{d}_y$ . So we only need to prove  $\Pr[\hat{d}_x < \hat{d}_y] \leq \Pr[d_x < \hat{d}_y]$  for any  $d_x, d_y$  satisfying  $d_x < \hat{d}_y$ .

According to the definition, we have

$$\begin{aligned} \Pr[\hat{d}_x < \hat{d}_y] &= \Pr[d_x + \eta_x < d_y + \eta_y] = \Pr[\eta_y > \eta_x + d_x - d_y] \\ &= \int_{-\infty}^{+\infty} \left( \int_{-\infty}^{\eta_y - d_x + d_y} \frac{\epsilon_x \epsilon_y}{4} e^{-(\epsilon_x |\eta_x| + \epsilon_y |\eta_y|)} d\eta_x \right) d\eta_y \end{aligned}$$

and

$$\begin{aligned} \Pr[d_x < \hat{d}_y] &= \Pr[d_x < d_y + \eta_y] = \Pr[\eta_y > d_x - d_y] \\ &= \int_{d_x - d_y}^{+\infty} \frac{\epsilon_y}{2} e^{-\epsilon_y |\eta_y|} d\eta_y. \end{aligned}$$

Let  $s = d_y - d_x$ . Let  $F : s \rightarrow \Pr[\hat{d}_x < \hat{d}_y]$  and  $G : s \rightarrow \Pr[d_x < \hat{d}_y]$ . From the definition, we know  $s > 0$ ,  $\lim_{s \rightarrow 0} F(s) =$

$\lim_{s \rightarrow 0} G(s) = \frac{1}{2}$  and  $\lim_{s \rightarrow +\infty} F(s) = \lim_{s \rightarrow +\infty} G(s) = 1$ . And we have

$$\begin{aligned} \frac{\partial F(s)}{\partial s} &= \frac{\epsilon_x \epsilon_y}{4} \left( \frac{e^{-s\epsilon_x} + e^{-s\epsilon_y}}{\epsilon_x + \epsilon_y} - \frac{e^{-s\epsilon_x} - e^{-s\epsilon_y}}{\epsilon_x - \epsilon_y} \right) \\ &= \frac{\epsilon_x \epsilon_y}{2} \cdot \frac{e^{-s\epsilon_y} \epsilon_x - e^{-s\epsilon_x} \epsilon_y}{(\epsilon_x + \epsilon_y)(\epsilon_x - \epsilon_y)} > 0, \\ \frac{\partial G(s)}{\partial s} &= \frac{\epsilon_y}{2} e^{-s\epsilon_y} > 0, \\ \frac{\partial F(s)}{\partial s} / \frac{\partial G(s)}{\partial s} &= \frac{\epsilon_x (\epsilon_x - e^{s(\epsilon_y - \epsilon_x)} \epsilon_y)}{(\epsilon_x + \epsilon_y)(\epsilon_x - \epsilon_y)}. \end{aligned}$$

Let  $\frac{\partial F(s)}{\partial s} / \frac{\partial G(s)}{\partial s} \leq 1$ . Then we have  $s \leq \frac{1}{\epsilon_x - \epsilon_y} \ln \frac{\epsilon_x}{\epsilon_y}$ . Let  $\frac{\partial F(s)}{\partial s} / \frac{\partial G(s)}{\partial s} \geq 1$ . Then we have  $s \geq \frac{1}{\epsilon_x - \epsilon_y} \ln \frac{\epsilon_x}{\epsilon_y}$ . That is to say  $\frac{\partial G(s)}{\partial s} \geq \frac{\partial F(s)}{\partial s}$  for  $s \in (0, \frac{1}{\epsilon_x - \epsilon_y} \ln \frac{\epsilon_x}{\epsilon_y})$  and  $\frac{\partial G(s)}{\partial s} \leq \frac{\partial F(s)}{\partial s}$  for  $s \in (\frac{1}{\epsilon_x - \epsilon_y} \ln \frac{\epsilon_x}{\epsilon_y}, +\infty)$ . According to Lemma IX.2, we have  $F(s) \leq G(s)$  for  $s \in (0, +\infty)$ .  $\square$

### B. Proof for Theorem VI.3

*Proof.* Let  $\hat{U}(st)$  be the overall utility of the strategy  $st$  with (i.e.,  $\hat{U}(st) = \Phi(st)$ ). Besides, we note the global optimal strategy as  $\hat{st}$ , the strategy of achieving best competing utility value as  $st^*$  and the worst competing utility value as  $st^\#$ . Then we have  $\hat{U}(\hat{st}) = \Phi(\hat{st})$ ,  $\hat{U}(st^*) = \Phi(st^*)$  and  $\hat{U}(st^\#) = \Phi(st^\#)$ . Thus,

$$EPoS = \frac{E(\hat{U}(st^*))}{E(OPT)} = \frac{E(\hat{U}(st^*))}{E(\hat{U}(\hat{st}))} \leq 1.$$

If we get the lower bound of  $E(\hat{U}(st^\#))$  and upper bound of  $E(\hat{U}(st))$ , then we can get the value of EPoA. As for  $E(\hat{U}(st^\#))$ , we have

$$\begin{aligned} E(\hat{U}(st^\#)) &\geq \min_k \sum_{t_i \in \mathcal{T}} \sum_{w_j \in \mathcal{W}} (s_{i,j}^{(k)} \cdot (v_i - f_d(d_{i,j})) - f_p(b_{i,j}^{(k)} \cdot \epsilon_{i,j})) \\ &\geq \sum_{t_i \in \mathcal{T}} \min_{R_j \ni t_i, U_j^L(i) > 0} U_j^L(i) = \sum_{t_i \in \mathcal{T}} U_{min}^+(i) \end{aligned}$$

As for  $E(\hat{U}(st))$ , we have

$$\begin{aligned} E(\hat{U}(st)) &\leq OPT \left( \sum_{t_i \in \mathcal{T}} \sum_{w_j \in \mathcal{W}} (s_{i,j} \cdot (v_i - f_d(\hat{d}_{i,j})) - f_p(b_{i,j} \cdot \epsilon_{i,j})) \right) \\ &\leq \sum_{t_i \in \mathcal{T}} \max_{R_j \ni t_i} U_j^H(i) = \sum_{t_i \in \mathcal{T}} U_{max}^+(i) \end{aligned}$$

So we have

$$EPoA = \frac{E(\hat{U}(st^*))}{E(OPT)} \geq \frac{\sum_{t_i \in \mathcal{T}} U_{min}^+(i)}{\sum_{t_i \in \mathcal{T}} U_{max}^+(i)}$$

$\square$

### C. Proof for Theorem VI.4

*Proof.* Let  $\mathcal{A}_j$  be the mechanism PGT applying to  $w_j$  with query  $f$  defined above. Let  $X_j$  be the location of  $w_j$ . For query  $\hat{f}(X_j) = [d_{i1,j}, \dots, d_{i|R_j|,j}]$ , we extend it to an equivalent query  $\hat{f}(X_j) = f(X_j) \cdot \mathcal{J}$ , where

$$\mathcal{J} = \begin{bmatrix} CP(b_{i1,j}) & & & \\ & CP(b_{i2,j}) & & \\ & & \ddots & \\ & & & CP(b_{i|R_j|,j}) \end{bmatrix}$$

is a block diagonal matrix. Actually,  $\hat{f}(X_j)$  means query  $d_{i_u,j}$  for  $\text{sum}(b_{i_u,j})$  times for  $u \in [|R_j|]$ . We denote the size of  $\hat{f}(X_j)$  as  $|\hat{f}|$  and the  $a$ -th element of  $\hat{f}(X_j)$  as  $\hat{f}(X_j)_a$ .

Let  $Y_j$  denote the set of all published obfuscated distances of the worker  $w_j$  to tasks in  $R_j$ . Then we have  $Y_j = \hat{f}(X_j) + [\eta_1, \eta_2, \dots, \eta_{|\hat{f}|}]$ , where  $\eta_a (1 \leq a \leq |\hat{f}|)$  is an i.i.d random variable drawn from  $Lap(1/\epsilon_a)$ . Hence we have

$$\begin{aligned} \frac{\Pr[\mathcal{A}_j(X_j) = Y_j]}{\Pr[\mathcal{A}_j(X'_j) = Y_j]} &= \prod_{a \in [|\hat{f}|]} \left( \frac{\exp(-\epsilon_a |Y_{j,a} - \hat{f}(X_j)_a|)}{\exp(-\epsilon_a |Y_{j,a} - \hat{f}(X'_j)_a|)} \right) \\ &= \prod_{t_i \in R_j} \prod_{u \in [sum(\mathbf{b}_{i,j})]} \left( \frac{\exp(-\epsilon_{i,j}^{(u)} |\tilde{d}_{i,j}^{(u)} - d_{i,j}|)}{\exp(-\epsilon_{i,j}^{(u)} |\tilde{d}'_{i,j}^{(u)} - d'_{i,j}|)} \right) \\ &\leq \prod_{t_i \in R_j} \prod_{u \in [sum(\mathbf{b}_{i,j})]} (\exp(\epsilon_{i,j}^{(u)} (|d_{i,j} - d'_{i,j}|))) \\ &= \prod_{t_i \in R_j} \exp(\mathbf{b}_{i,j} \epsilon_{i,j} (|d_{i,j} - d'_{i,j}|)) \\ &\leq \exp\left(\sum_{t_i \in R_j} \mathbf{b}_{i,j} \epsilon_{i,j} r_j\right). \end{aligned}$$

Because  $X_j$  contains only one element, then we have PGT satisfies  $(\sum_{t_i \in R_j} \mathbf{b}_{i,j} \epsilon_{i,j} r_j)$ -local differential privacy for each worker  $w_j$ .  $\square$

#### D. Experiment Result for the Uniform Data set

The experiment result of the uniform data set is shown in this section.

The time cost is shown in Figure 18.

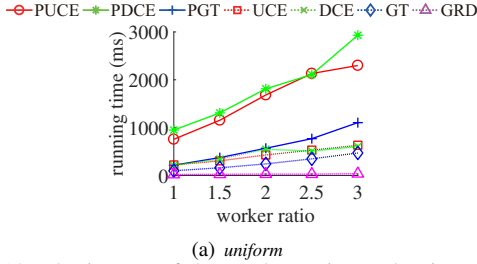


Fig. 18: The impact of the worker ratio on the time cost.

The impact of task value on utility is shown in Figure 19.

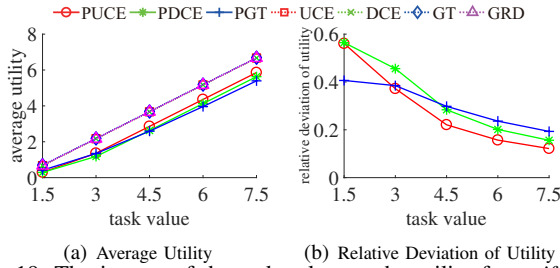


Fig. 19: The impact of the task value on the utility for *uniform*.

The impact of worker range on utility is shown in Figure 20. The impact of worker ratio on utility is shown in Figure 21. The impact of task value on travel distance is shown in Figure 22.

The impact of worker range on travel distance is shown in Figure 23.

The impact of worker ratio on travel distance is shown in Figure 24.

The impact of worker ratio on PPCF and non-PPCF is shown in Figure 25.

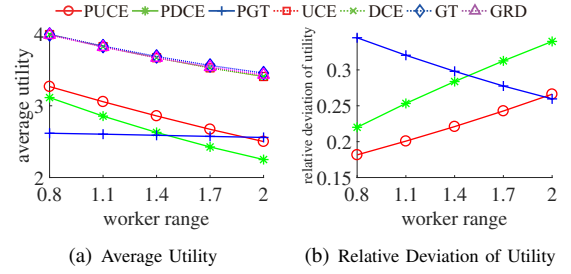


Fig. 20: The impact of the worker range on the utility for *uniform*.

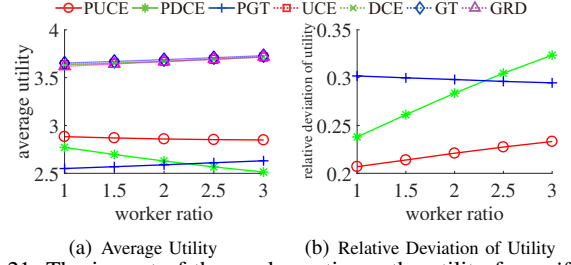


Fig. 21: The impact of the worker ratio on the utility for *uniform*.

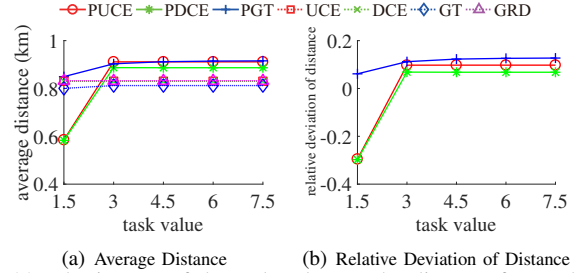


Fig. 22: The impact of the task value on the distance for *uniform*.

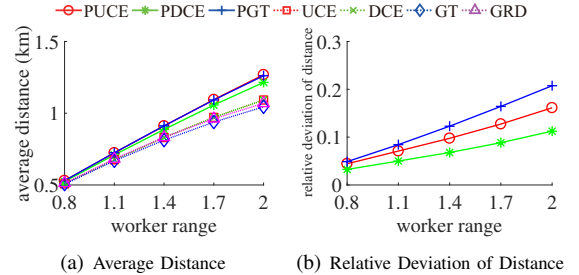


Fig. 23: The impact of the worker ratio on the distance for *uniform*.

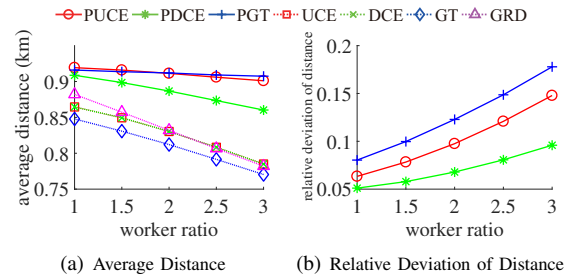


Fig. 24: The impact of the worker ratio on the distance for *uniform*.

#### E. Additional Definitions

**Definition 8 (One-to-one Match).** Let  $G = (U, E, V)$  be a bipartite graph, and  $M \subseteq E$  be a match in  $G$ .  $M$  is called a one-to-one match if for any two different edges  $e_{u,v}, e_{u',v'} \in M$ ,  $e_{u,v} \cap e_{u',v'} = \phi$ .

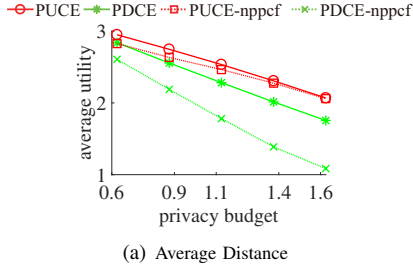


Fig. 25: The impact of privacy on the utility.

Let  $S_G(M) = [s_{u,v}]_{u \in U, v \in V}$  denote a state matrix of  $M$ . Here,  $s_{u,v} = 1$  when  $e_{u,v} \in M$ ,  $s_{u,v} = 1$ ; otherwise,  $s_{u,v} = 0$ .

We utilize Differential Privacy [17] to disturb the raw data and measure the privacy cost of workers' proposals for tasks.

**Definition 9** (Differential Privacy [17], DP). A randomized algorithm  $\mathcal{A}$  with domain  $\mathbb{N}^{|\mathcal{X}|}$  is  $(\epsilon, \delta)$ -differential private if for all  $S \subseteq \text{Range}(\mathcal{A})$  and for all  $x, y \in \mathbb{N}^{|\mathcal{X}|}$  such that  $\|x - y\|_1 \leq 1$ :

$$\Pr[\mathcal{A}(x) \in S] \leq \exp(\epsilon) \Pr[\mathcal{A}(y) \in S] + \delta,$$

where  $\|\cdot\|_1$  is the  $\ell_1$  norm of an vector,  $\Pr[\cdot]$  denotes the probability of an event. Especially, when  $\delta = 0$ ,  $\mathcal{A}$  is  $\epsilon$ -differential private.

When  $x$  and  $y$  consists of single element,  $\mathcal{S}$  is also called a local randomizer, which provides *local differential privacy* (LDP) guarantees [34].

The Laplace mechanism [17] is the most well know perturbation methods for numeric values that satisfy the definition of differential privacy. Given a function  $f$  outputting a numeric value vector  $f(\cdot)$ , the laplace mechanism is able to transform  $f$  into a differentially private algorithm by adding random noise to each entry of  $f(\cdot)$ . The random noise is sampled from laplace distribution.

The scale of the random noise is relevant to the  $\ell_1$ -sensitivity of  $f$  as well as a predetermined privacy budget  $\epsilon$ . The  $\ell_1$ -sensitivity of  $f$  is defined as the maximum possible difference between any two vector  $x$  and  $y$  with their  $\ell_1$  distance as 1:

**Definition 10** ( $\ell_1$ -sensitivity [17]). The  $\ell_1$ -sensitivity of a function  $f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^k$  is:

$$\Delta f = \max_{x, y \in \mathbb{N}^{|\mathcal{X}|}, \|x - y\|_1 = 1} \|f(x) - f(y)\|_1.$$

Thus, the laplace mechanism is stated as follows:

**Definition 11** (The Laplace Mechanism [17]). Given any function  $f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^k$ , the Laplace mechanism is defined as:

$$\mathcal{A}_L(x, f(\cdot), \epsilon) = f(x) + (Y_1, \dots, Y_k)$$

where  $Y_i$  ( $i \in [k]$ ) is an i.i.d. random variable drawn from  $Lap(\Delta f / \epsilon)$ .

In order to compare two disturbed value conveniently, we introduce the Probability Compare Function [3] in Definition 6.