# Location Privacy Protection in Vehicle-Based Spatial Crowdsourcing via Geo-Indistinguishability

Chenxi Qiu
*Department of Computer Science*
*Rowan University*
*Email: qiu@rowan.edu*

Anna Cinzia Squicciarini
*College of Information Science and Technology*
*The Pennsylvania State University*
*Email: acs20@psu.edu*

***Abstract*—Nowadays, vehicles have been increasingly adopted as participants in many spatial crowdsourcing (SC) applications. Similar to other SC applications, location privacy is of great concern to vehicle workers as they are required to disclose their own location information to servers to facilitate the utilities of SC services. Traditional location privacy protection mechanisms cannot be directly applied to vehicle-based SC since they assume workers' location on a 2-dimensional plane, which does not take into account the features of vehicle workers' mobility in vehicle road networks. Accordingly, in this paper, we aim at addressing issues related to Vehicle-based spatial crowdsourcing Location Privacy (VLP) in vehicle road networks. Our objective is to design a location obfuscation strategy to minimize the loss of quality-of-service (QoS) due to task distribution with location obfuscation, while guaranteeing *geo-indistinguishability* to be satisfied. Considering the computational complexity of the VLP problem, by resorting to discretization, we approximate VLP to a linear programming problem that can be solved by existing well-developed approaches (such as the simplex method). To further improve the time efficiency, we reduce the number of constraints in VLP by exploiting key features of geo-indistinguishability in vechicle road networks (such as *transitivity*). Finally, our experimental results demonstrate that our approach can achieve a reasonable approximation of the minimum QoS loss with location privacy protected, and also outperforms a known state-of-the-art location obfuscation strategy in terms of both QoS and privacy.**

***Keywords*-Location privacy; spatial crowdsourcing; geo-indistinguishability;**

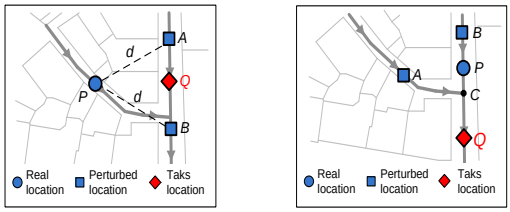
## I. Introduction

*Spatial Crowdsourcing (SC)* [1] has emerged as a new mode of crowdsourcing to enable requesters to outsource their *spatial tasks* (i.e., tasks related to a location) to a set of mobile workers. In SC, task requesters register through a centralized *server* and publish tasks with target locations or spatial routes. If a worker accepts the tasks, he/she needs to physically travel to the tasks' location to perform the tasks. In the last few years, SC has been applied to many different domains, such as smart cities [2] and environmental sensing [3]. Particularly, with the advent of intelligent transport system, *vehicle-based spatial crowdsourcing (VSC)* is evolving rapidly [4]. For example, many recent studies have proposed to use vehicle crowdsourcing workers as mobile agents to help maintain vehicle ad hoc networks (VANETs), for tasks such as data dissemination and query processing (e.g., [4]–[6]). In some other vehicle-related applications, VSC has been used for data sharing and collection [7], or to improve traditional transportation systems such as Uber [4].

To ensure that tasks are completed in a timely fashion and vehicle workers' *traveling* is cost-effective, usually, a server in VSC matches available workers with tasks that have the *shortest path distance (ShPD)* to the workers [8]–[10]. To this extent, workers are required to disclose their locations to the server in real time. Such location information exposure, however, may lead to privacy breaches not only related to the whereabouts of a vehicle, but also related to other sensitive information, e.g., home, sexual preferences, religious inclinations, etc [11].

In fact, location privacy research has gained great attention over the last decade, and a variety of location privacy protection strategies have been developed, such as *k-anonymity* [12]–[14], *cloaking* [15], [16], and *pseudonym* based methods [17]. Particularly, the *location obfuscation* approaches, which allow users to generate and report dummy locations that look equally likely to be the true location, have been widely used for protecting location privacy due to their high efficiency and effectiveness in various applications [18]–[26].

As a growing effort aims to address the location privacy issues via location obfuscation, a formal notion of location privacy, namely *Geo-Indistinguishability* (or *Geo-I*), was introduced by Andrés et al. recently [27]. Geo-I can be considered as a generalization of the statistical notion *differential privacy* [28]. According to Geo-I, if two locations are geographically close, they will have similar probabilities to generate a certain reported location. In the other words, the reported location will not provide enough information to an adversary to distinguish the true location among nearby ones. Subsequent to this notion of Geo-I, a variety of improved location obfuscation methods have been proposed [23]–[26]. Particularly, considering that users' location privacy is usually protected at the expense of *quality of service (QoS)*, some of these works introduce optimization-based approaches (e.g., linear programming) to minimize the QoS loss and still preserve privacy [23].




(a) Example I (b) Example II

Figure 1. An example of location obfuscation on vehicle road map.

In most of these approaches, both QoS loss and privacy metrics are defined to be positively correlated to the Euclidean distance between the true location and obfuscated location [18], [19], [23]–[25], [27]. Such measure can be applied to scenarios wherein workers are able to move freely on a 2-dimensional (2-D) plane, where the shortest path distance between any pair of points is defined as their Euclidean distance. However, compared with traditional mobile workers (i.e., human beings), workers' mobility in VSC is more structured since vehicles have to operate on a *vehicle road network (or road network)* and usually cannot move to their destination on a straight line. This, in many cases, leads to a significant different sensitivity of QoS loss to location obfuscation. Figure 1(a)(b) provides two examples, where in both figures, $P$ represents the vehicle's true location, $A$ and $B$ represent two reported locations after obfuscation, whereas $Q$ represents the task location:

**Example I** (Figure 1(a)): $A$ and $B$ have the same Euclidean distance to both $P$ and $Q$, indicating that, on a 2D plane, the two reported location points introduce the same estimation error of path distance (or QoS loss) from $P$ to $Q$. However, on the road map, being reported at $A$ or $B$ is different: both $B$ and $P$ need to take a detour to reach $Q$, while $A$ can reach $Q$ almost with a straight line. Hence, the QoS loss generated by $A$ is much higher than that of $B$.

**Example II** (Figure 1(b)): Compared with $B$, $A$ has a longer Euclidean distance to $P$ but shorter Euclidean distance to $Q$. Hence, the two reported location points offer different QoS loss in 2D plane. However, since $A$ and $B$ have the same path distance to $Q$, the QoS loss of the two points on the vehicle road map will be the same.

Considering the different features of vehicle workers' mobility in a road network than in a 2D plane, in this paper, we aim to solve the *Vehicle based spatial Crowdsourcing Location Privacy (VLP)* problem in a *road map*. More precisely, we model the road map by a *weighted directed graph* and assume that both workers' and tasks' locations are the points on the graph. We consider a location obfuscation approach under which each worker is allowed to report an obfuscated location instead of his/her true location and the obfuscated location is probabilistically distributed over the graph. Our objective is to determine the obfuscated (or reported) location's probability distribution over the graph (so-called *location obfuscation strategy*), such that *1) the QoS loss is minimized and 2) Geo-I is satisfied*. Specifically, for each vehicle, we define QoS loss as the *expected estimation error of the shortest path distance* from the vehicle to all the tasks. As for privacy, instead of adopting the Geo-I defined in [27], which is Euclidean distance based, we redefine the notion of Geo-I based on path distance in directed weighted graphs (details can be found in Definition 2.1). This correction, however, increases the complexity of VLP from an algorithmic perspective.

We start by discussing VLP in a general case, where the probability distribution of a reported location is a general function defined over the whole graph. Considering the problem's computational intractability, we approximate VLP via discretization: Each edge in the graph is partitioned into small intervals and the locations within each interval don't need to be differentiated. The approximated VLP, called *Discretized VLP* or *D-VLP*, can be then formulated as a *linear programming* problem. For theoretical interests, we also derive a lower bound for VLP. By comparing this lower bound with the optimal solution of D-VLP, we can check the gap between the optimal solutions of D-VLP and VLP.

Note that Geo-I, as a privacy requirement, generates $O(K^3)$ constraints in D-VLP ($K$ denotes the amount of road intervals partitioned in D-VLP), leading to a high computation cost in linear programming [29]. Fortunately, by exploiting important features of Geo-I in road network (e.g., *transitivity* described in Property 3.2), we prove that, instead of constraining all pairs of intervals by Geo-I, constraining adjacent intervals in the graph will be sufficient to keep the optimality, which significantly improves the time efficiency of our approach.

With respect to performance, results based on a dataset of taxi cabs' trajectory in Rome, Italy [30] demonstrate that our approach outperforms a state-of-the-art location obfuscation mechanism [23] in terms of both QoS loss and privacy (e.g., on average reduce the QoS loss by 12.35% and increase the expected error from adversary by 6.91%). In addition, we show that D-VLP offers a reasonably good approximation with VLP in term of the QoS loss. Finally, we compare the computation time for solving D-VLP with and without our proposed constraint reduction method, from which we demonstrate our approach can significantly reduce the number of constraints led by Geo-I and hence improve the time efficiency for D-VLP (on average by 87.9%).

In summary, our contributions include:

1) We formulate a new problem called the *VSC Location Privacy (VLP)* problem, of which our objective is to minimize the QoS loss of VSC without compromising workers' location privacy. We start discussing VLP in a general case and provide a theoretical lower bound of the QoS loss in VLP.

2) Considering the computational intractability of VLP,

we approximate VLP to a linear programming problem by discretization. We then propose a constraint reduction approach to further improve the time efficiency for solving the LP.

3) We conduct a simulation based on real trace to test the performance our location obfuscation approach. The simulation results demonstrate that our approach can achieve the optimal QoS closely, and also outperforms one existing 2D plane-based method in terms of both QoS and privacy.

The remainder of the paper is organized as follows: We introduce the model and the VLP problem in Section II and propose a time efficient solution in Section III. In Section IV, we test the performance of our approach. Finally, we present related work in Section V and conclude in Section VI.

## II. Model and Problem Formulation

In this section, we first introduce the model, including notations and assumptions that will be used throughout the paper in Section II-A. Based on the model, we then formally formulate the VLP problem in Section II-B.

### A. Model

We consider a scenario where a *server* needs to estimate the *shortest path distance (ShPD)* from a *vehicle worker* (or *worker*) to a spatial task with the location specified in a *road network*. Like [31], we can represent the road network by a set of roads. When a road intersects, furcates, joins with other roads, or turns into a different direction, a connection is created (as shown in Figure 2). These connections divide roads into multiple *road segments*, which only connect with other road segments at their end points. Accordingly, the road network can be represented by a *weighted directed graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the *connection set* and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the *route segment (edge) set*. Each edge $e \in \mathcal{E}$ is directed, presented by an ordered pair $(v_e^{\mathrm{s}}, v_e^{\mathrm{e}})$ $(v_e^{\mathrm{s}}, v_e^{\mathrm{e}} \in \mathcal{V})$, where $v_e^{\mathrm{s}}$ and $v_e^{\mathrm{e}}$ denote the starting and the ending connections of $e$, respectively. That is, vehicles can only move from $v_e^{\mathrm{s}}$ to $v_e^{\mathrm{e}}$ on $e$ in the road network. Each $e$ is allocated with a weight $w_e$ representing the path distance from $v_e^{\mathrm{s}}$ to $v_e^{\mathrm{e}}$.

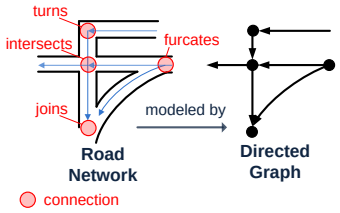


Figure 2. Edge partition.

To derive the ShPD between the worker and the task, besides the road network information and the task location, the server also requires the worker to report his/her own location in real time. We assume both task and worker are located in the road network $G$, and let $\mathbf{p}$ and $\mathbf{q}$ denote the worker and the task's location, respectively. Each location point $\mathbf{p}$ (or $\mathbf{q}$) is represented by a ordered pair: $\mathbf{p} = (e, x)$, where $e$ represents the edge that $\mathbf{p}$ is located in and $x$ ($x \in (0, w_e]$) denotes the path distance from $\mathbf{p}$ to $e$'s endpoint $v_e^{\mathrm{e}}$.

Given any pair of locations $\mathbf{v}$ and $\mathbf{v}'$ in the road network $\mathcal{G}$, we let $d_{\mathcal{G}}(\mathbf{v}, \mathbf{v}')$ represent the ShPD from $\mathbf{v}$ to $\mathbf{v}'$ in $\mathcal{G}$ (in one direction). Note that $d_{\mathcal{G}}(\mathbf{v}, \mathbf{v}')$ and $d_{\mathcal{G}}(\mathbf{v}', \mathbf{v})$ are possibly different since they measure the traveling distance of different paths. We let $d_{\mathcal{G}}^{\min}(\mathbf{v}, \mathbf{v}')$ denote the ShPD between $\mathbf{v}$ and $\mathbf{v}'$ (in two directions):

$$d_{\mathcal{G}}^{\min}(\mathbf{v}, \mathbf{v}') = \min\{d_{\mathcal{G}}(\mathbf{v}, \mathbf{v}'), d_{\mathcal{G}}(\mathbf{v}', \mathbf{v})\}. \quad (1)$$

Table I lists the main notations and their descriptions used throughout this paper.

Table I
Main Notations and Definition

| Notation | Description |
|---|---|
| $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ | The road network, where $\mathcal{V}$ and $\mathcal{E}$ denote its connection set and its edge set |
| $w_e$ | The weight (length) of edge $e$ |
| $\mathbf{p}$ | The worker's true location |
| $\tilde{\mathbf{p}}$ | The worker's obfuscated location |
| $\mathbf{q}$ | The task location |
| $v_e^{\mathrm{s}}$ ($v_e^{\mathrm{e}}$) | The starting (ending) point of edge $e$ |
| $e(\mathbf{p})$ ($e(\mathbf{q})$) | The edge that $\mathbf{p}$ ($\mathbf{q}$) is positioned in |
| $d_{\mathcal{G}}(\mathbf{v}, \mathbf{v}')$ | The ShPD from location $\mathbf{v}$ to location $\mathbf{v}'$ in $\mathcal{G}$ (in one direction) |
| $d_{\mathcal{G}}^{\min}(\mathbf{v}, \mathbf{v}')$ | The ShTD between location $\mathbf{v}$ and location $\mathbf{v}'$ in $\mathcal{G}$ (in two directions) |
| $\Delta d_{\mathcal{G}}(\mathbf{p}, \tilde{\mathbf{p}}; \mathbf{q})$ | The difference between $d_{\mathcal{G}}(\mathbf{p}, \mathbf{q})$ and $d_{\mathcal{G}}(\tilde{\mathbf{p}}, \mathbf{q})$ |
| $\Delta \underline{d}_{\mathcal{G}}(\mathbf{p}, \tilde{\mathbf{p}}; \mathbf{q})$ | A lower bound of $\Delta d_{\mathcal{G}}(\mathbf{p}, \tilde{\mathbf{p}}; \mathbf{q})$ defined by Equations (31)–(34) |
| $f_P(\mathbf{p})$ ($f_Q(\mathbf{q})$) | The prior PDF of $\mathbf{p}$ ($\mathbf{q}$) |
| $f_{\tilde{P}}(\tilde{\mathbf{p}}\|P = \mathbf{p})$ | The conditional PDF of the obfuscated location $\tilde{\mathbf{p}}$ given the true location $\mathbf{p}$ |

**Threat Model**: **a) Worker**. We assume an untrusted server (e.g., a remote server in the cloud [10]), which information can be possibly disclosed or leaked to an adversary. To maintain location privacy, the worker will report an obfuscated location $\tilde{\mathbf{p}}$ instead of his/her true location $\mathbf{p}$, where $\tilde{\mathbf{p}}$ is probabilistically determined. We use random variables $P$ and $\tilde{P}$ to represent the worker's true and obfuscated location, respectively. When reporting the obfuscated location, the worker's true location is given, i.e., $P = \mathbf{p}$, and hence the reported location distribution can be described by a conditional PDF $f_{\tilde{P}}(\tilde{\mathbf{p}}|P = \mathbf{p})$, where $\sum_{e \in \mathcal{E}} \int_{[0, w_e]} f_{\tilde{P}}(\tilde{\mathbf{p}}|P = \mathbf{p}) \mathrm{d}x = 1$. The *obfuscation strategy* of the worker is essentially the collection of conditional PDFs given all possible $\mathbf{p}$

$$\mathcal{F} = \{f_{\tilde{P}}(\tilde{\mathbf{p}}|P = \mathbf{p}) | \mathbf{p} = (e, x), e \in \mathcal{E}, x \in (0, w_e]\}. \quad (2)$$

**b) Adversary**. The location of the targeted worker from the adversary is assumed to be estimated by a probabilistic model. More precisely, given a reported location $\tilde{\mathbf{p}}$ from the worker, the adversary tries to find the true location $\mathbf{p}$ by calculating $\mathbf{p}$'s probability distribution. Here, we consider the worst-case scenario, where the adversary has full information about the worker's obfuscation strategy $\mathcal{F}$ and the worker's prior PDF $f_P(\mathbf{p})$. Then, given the reported $\tilde{\mathbf{p}}$, the adversary can derive the PDF of the true location $\mathbf{p}$ by the Bayes' Theorem [32]:

$$f_P\left(\mathbf{p}|\tilde{P}=\tilde{\mathbf{p}}\right)=\frac{f_{\tilde{P}}\left(\tilde{\mathbf{p}}|P=\mathbf{p}\right)f_P\left(\mathbf{p}\right)}{\int_{\mathcal{G}} f_{\tilde{P}}\left(\tilde{\mathbf{p}}|P=\mathbf{p}'\right)f_P\left(\mathbf{p}'\right)\mathrm{d}\mathbf{p}'} \tag{3}$$

### *B. Problem Formulation*

Before formulating the problem, we first define the two metrics that are considered for the location obfuscation strategy: *privacy* and *QoS loss*.

*1) Privacy:* We aim to achieve *quasi-indistinguishability* or *Geo-Indistinguishability* (*Geo-I*) [27] for any pair of locations that are close to each other. Geo-I corresponds to a generalized version of the well-known concept of *differential privacy*. The idea of Geo-I on a 2D plane is to require a small change of a single user's location, measured by Euclidean distance, so as not to affect the distribution of his/her reported location too much. Following this idea, we redefine Geo-I on a weighted directed graph in Definition 2.1, where we measure the difference between any pair of locations by their ShPD on graph, rather than their Euclidean distance. Particularly, for each pair of locations, we consider the ShPD in both directions and pick up the shorter one as the measure of privacy.

**Definition 2.1:** A location obfuscation strategy satisfies $(\epsilon, r)$-Geo-I if and only if for any pair of true locations $\mathbf{p}_i$ and $\mathbf{p}_l$ such that $d_{\mathcal{G}}^{\min}(\mathbf{p}_i,\mathbf{p}_l)\leq r$ and for any obfuscated location $\tilde{\mathbf{p}}$,

$$\frac{f_P\left(\mathbf{p}_i|\tilde{P}=\tilde{\mathbf{p}}\right)}{f_P\left(\mathbf{p}_l|\tilde{P}=\tilde{\mathbf{p}}\right)}\leq e^{\epsilon d_{\mathcal{G}}^{\min}(\mathbf{p}_i,\mathbf{p}_l)}\frac{f_P\left(\mathbf{p}_i\right)}{f_P\left(\mathbf{p}_l\right)}, \tag{4}$$

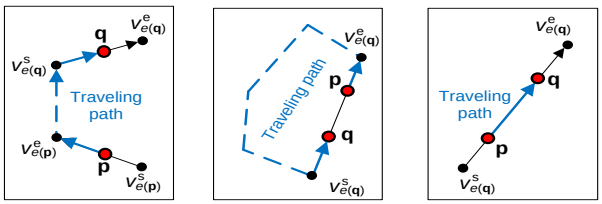
where $r$ is the radius of the obfuscation area and $\epsilon$ is the parameter to quantify how much information of the true location will be disclosed according to the reported location, where higher $\epsilon$ implies more information to be disclosed.

According to Equation (3), Equation (4) can be rewritten as

$$f_{\tilde{P}}\left(\tilde{\mathbf{p}}|P=\mathbf{p}_i\right)\leq e^{\epsilon d_{\mathcal{G}}^{\min}(\mathbf{p}_i,\mathbf{p}_l)}f_{\tilde{P}}\left(\tilde{\mathbf{p}}|P=\mathbf{p}_l\right). \tag{5}$$

*2) QoS loss:* Given the worker's obfuscated location $\tilde{\mathbf{p}}$, his/her true location $\mathbf{p}$, and the task location $\mathbf{q}$, we measure the QoS loss by the *estimation error of ShPD* to the task location $\mathbf{q}$, which, more precisely, is defined as the difference between the estimated ShPD $d_{\mathcal{G}}(\tilde{\mathbf{p}},\mathbf{q})$ and the true ShPD $d_{\mathcal{G}}(\mathbf{p},\mathbf{q})$:

$$\Delta d_{\mathcal{G}}\left(\mathbf{p},\tilde{\mathbf{p}};\mathbf{q}\right)=\left|d_{\mathcal{G}}(\mathbf{p},\mathbf{q})-d_{\mathcal{G}}(\tilde{\mathbf{p}},\mathbf{q})\right|. \tag{6}$$




(a) C1 (b) C2 (c) C3

Figure 3. Derivation of $d_{\mathcal{G}}(\mathbf{p},\mathbf{q})$ under three cases.

We now analyze how the obfuscated location will affect the accuracy of estimated ShPD. As opposed to 2D plane, the QoS loss in a vehicle road map is highly impacted by the topology of the network. Here, we first derive $d_{\mathcal{G}}(\mathbf{p},\mathbf{q})$ by considering the following two cases (we use $e(\mathbf{q})$ and $e(\mathbf{p})$ to represent the edges that $\mathbf{q}$ and $\mathbf{p}$ are located in)

C1 1) When $e(\mathbf{q})\neq e(\mathbf{p})$ (Figure 3(a)), or
2) When $e(\mathbf{q})=e(\mathbf{p})$ but $\mathbf{p}$ has shorter ShPD to its endpoint of $e(\mathbf{p})$, $v^{\mathrm{e}}_{e(\mathbf{p})}$, than $\mathbf{q}$ (Figure 3(b)):
In this case, the worker's traveling path has to first reach the current edge's endpoint $v^{\mathrm{e}}_{e(\mathbf{p})}$, then the starting point of $\mathbf{q}$'s edge, $v^{\mathrm{s}}_{e(\mathbf{q})}$, and finally the destination location $\mathbf{q}$. Hence, the ShPD from $\mathbf{p}$ to $\mathbf{q}$ is calculated by

$$\begin{aligned} & d_{\mathcal{G}}(\mathbf{p},\mathbf{q}) \\ = \; & d_{\mathcal{G}}\left(\mathbf{p},v^{\mathrm{e}}_{e(\mathbf{p})}\right)+d_{\mathcal{G}}\left(v^{\mathrm{e}}_{e(\mathbf{p})},v^{\mathrm{s}}_{e(\mathbf{q})}\right)+d_{\mathcal{G}}\left(v^{\mathrm{s}}_{e(\mathbf{q})},\mathbf{q}\right) \\ = \; & d_{\mathcal{G}}\left(v^{\mathrm{e}}_{e(\mathbf{p})},v^{\mathrm{s}}_{e(\mathbf{q})}\right)+x_{\mathbf{p}}+l_{e(\mathbf{q})}-x_{\mathbf{q}}. \end{aligned} \tag{7}$$

C2 When $e(\mathbf{q})=e(\mathbf{p})$ and $\mathbf{p}$ has longer path distance to the edge's endpoint $v^{\mathrm{e}}_{e(\mathbf{p})}$ than $\mathbf{q}$ (Figure 3(c)), the ShPD from $\mathbf{p}$ to $\mathbf{q}$ is

$$d_{\mathcal{G}}\left(\mathbf{p},\mathbf{q}\right) = d_{\mathcal{G}}\left(\mathbf{p},v^{\mathrm{e}}_{e(\mathbf{p})}\right)-d_{\mathcal{G}}\left(\mathbf{q},v^{\mathrm{e}}_{e(\mathbf{p})}\right) \tag{8}$$

$$= x_{\mathbf{p}}-x_{\mathbf{q}} \tag{9}$$

A similar derivation can be applied to $d_{\mathcal{G}}(\tilde{\mathbf{p}},\mathbf{q})$. Considering the possible $(\mathbf{p},\mathbf{q})$ (and $(\tilde{\mathbf{p}},\mathbf{q})$) in the above two cases, we can derive $\Delta d_{\mathcal{G}}\left(\mathbf{p},\tilde{\mathbf{p}};\mathbf{q}\right)$ as

$$\begin{aligned} & \Delta d_{\mathcal{G}}\left(\mathbf{p},\tilde{\mathbf{p}};\mathbf{q}\right) \\ = \; & \begin{cases} \left|d_{\mathcal{G}}(v^{\mathrm{e}}_{e(\mathbf{p})},v^{\mathrm{s}}_{e(\mathbf{q})})-d_{\mathcal{G}}(v^{\mathrm{e}}_{e(\tilde{\mathbf{p}})},v^{\mathrm{s}}_{e(\mathbf{q})})+x_{\mathbf{p}}-x_{\tilde{\mathbf{p}}}\right| & C(1,1) \\ \left|d_{\mathcal{G}}(v^{\mathrm{e}}_{e(\mathbf{p})},v^{\mathrm{s}}_{e(\mathbf{q})})+x_{\mathbf{p}}+l_{e(\mathbf{q})}-x_{\tilde{\mathbf{p}}}\right| & C(1,2) \\ \left|d_{\mathcal{G}}(v^{\mathrm{e}}_{e(\tilde{\mathbf{p}})},v^{\mathrm{s}}_{e(\mathbf{q})})+x_{\tilde{\mathbf{p}}}+l_{e(\mathbf{q})}-x_{\mathbf{p}}\right| & C(2,1) \\ \left|x_{\tilde{\mathbf{p}}}-x_{\mathbf{p}}\right| & C(2,2) \end{cases}, \end{aligned} \tag{10}$$

where $C(i,j)$ represents when $(\mathbf{p},\mathbf{q})$ is in case $i$ and $(\tilde{\mathbf{p}},\mathbf{q})$ is in case $j$.

In addition, we assume that the task location $\mathbf{q}$ won't be exposed to the worker before the worker selects his/her obfuscated location. The worker however has the prior distribution of the task, $f_Q(\mathbf{q})$, based on the historical record, where $Q$ denotes the random variable to describe the task location. Then, given the location obfuscation approach $\mathcal{F}$,

the worker can obtain its expected estimation error of ShPD (QoS loss):

$$\mathbf{E}\left(\Delta d_{\mathcal{G}}\left(P,\tilde{P};Q\right)\right) \tag{11}$$
$$= \int\int\int \Delta d_{\mathcal{G}}\left(\mathbf{p},\tilde{\mathbf{p}};\mathbf{q}\right) f_Q(\mathbf{q}) f_P(\mathbf{p}) f_{\tilde{P}}(\tilde{\mathbf{p}}|P=\mathbf{p}) \mathrm{d}\tilde{\mathbf{p}}\mathrm{d}\mathbf{q}\mathrm{d}\mathbf{p}$$

**Problem Formulation**. Based on our definition of Geo-I (Equation (5)) and the QoS loss (Equation (11)), we formulate the *VSC Location privacy Protection (VLP)* problem as:

$$\min \quad \mathbf{E}\left(\Delta d_{\mathcal{G}}\left(P,\tilde{P};Q\right)\right) \tag{12}$$
$$\text{s.t.} \quad f_{\tilde{P}}\left(\tilde{\mathbf{p}}|P=\mathbf{p}_i\right) \le e^{\epsilon d_{\mathcal{G}}^{\min}(\mathbf{p}_i,\mathbf{p}_l)} f_{\tilde{P}}\left(\tilde{\mathbf{p}}|P=\mathbf{p}_l\right), \quad \forall \mathbf{p}_i,\mathbf{p}_l,\tilde{\mathbf{p}} \text{ in } \mathcal{G}, \text{ with } d_{\mathcal{G}}^{\min}(\mathbf{p}_i,\mathbf{p}_l) \le r. \tag{13}$$

The objective of VLP is to determine each location obfuscation strategy $f_{\tilde{P}}(\tilde{\mathbf{p}}|P=\mathbf{p})$ in $\mathcal{F}$ such that the estimation error of ShPD is minimized (Equation (12)) and $(\epsilon, r)$-Geo-I is satisfied (Equation (13)).

However, finding the optimal $f_{\tilde{P}}(\tilde{\mathbf{p}}|P=\mathbf{p})$ is computational intractable as it is difficult to describe $f_{\tilde{P}}(\tilde{\mathbf{p}}|P=\mathbf{p})$ (a general continuous function) with finite number of decision variables. As workers are mostly highly dynamic, and hence are required to update their location information in a timely fashion, it is of great importance to find a location obfuscation strategy that can *achieve near-minimum QoS loss with low time complexity*.

## III. Algorithm Design and Analysis

In this section, we aim to design a time efficient algorithm for VLP. The basic idea is to approximate VLP to a linear programming problem via discretization (Section III-A). After that, by exploring key features of Geo-I in road networks, we design an approach that can further reduce the complexity of the descretized VLP (Section III-B). Table lists the additional notations used in this section.

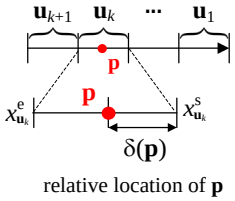



Figure 4. Edge partition.

### A. Problem Approximation

The obfuscated location distribution in VLP is a general function defined in a continuous region, i.e., the road network, which cannot be represented by finite number of decision variables. As a solution, we approximate VLP by discretization, in which we only need to consider the obfuscated location probability in intervals instead of in a continuous region. We denote the discretized problem by *discretized-VLP* or *D-VLP*. More precisely, we formulate D-VLP from VLP by the following three steps:

Table II
Additional notations and definition in Section III

| Notation | Description |
|---|---|
| $\mathbf{u}_k$ | The $k$th interval partitioned in $\mathcal{G}$ |
| $\mathbf{u}_k^{\mathrm{s}}$ ($\mathbf{u}_k^{\mathrm{e}}$) | The starting (ending) point of $\mathbf{u}_k$ |
| $\mathcal{U}$ | The set of intervals $\mathcal{U}=\{\mathbf{u}_1,...,\mathbf{u}_K$ |
| $\delta$ | The length of each partitioned interval $\mathbf{u}_k$ |
| $\delta(\mathbf{p})$ ($\delta(\tilde{\mathbf{p}})$) | The relative location of $\mathbf{p}$ ($\tilde{\mathbf{p}}$) |
| $\mathcal{G}'=(\mathcal{U}',\mathcal{E}')$ | The auxiliary graph describing the interval set $\mathcal{U}$, where $\mathcal{U}'$ corresponds $\mathcal{U}$ and $\mathcal{E}'$ corresponds the distance between adjacent intervals in $\mathcal{U}$ |
| $\mathbf{u}'_k$ | The vertex corresponding to $\mathbf{u}_k$ in $\mathcal{U}'$ |
| $\overset{G(\epsilon,r)}{\simeq}$ | $\mathbf{u}'_i \overset{G(\epsilon,r)}{\simeq} \mathbf{u}'_l$ if Geo-I is satisfied in the direction from $\mathbf{u}'_i$ to $\mathbf{u}'_l$ (Definition 3.2) |

**S1** Each edge is partitioned into route intervals with length $\delta$ (as depicted in Figure 4). We let $\mathcal{U}=\{\mathbf{u}_1,\mathbf{u}_2,...,\mathbf{u}_K\}$ ($K=|\mathcal{U}|$) denotes the set of intervals in the road network $\mathcal{G}$, and let $\mathbf{u}_k^{\mathrm{s}}=\left(e,x^{\mathrm{s}}_{\mathbf{u}_k}\right)$ and $\mathbf{u}_k^{\mathrm{e}}=\left(e,x^{\mathrm{e}}_{\mathbf{u}_k}\right)$ denote the two endpoints of each $\mathbf{u}_k$ (in edge $e$), where $x^{\mathrm{s}}_{\mathbf{u}_k}-x^{\mathrm{e}}_{\mathbf{u}_k}=\delta$. For a true location $\mathbf{p}=(e,x)$ that is in $\mathbf{u}_k$, we call $\delta(\mathbf{p})=x-x^{\mathrm{e}}_{\mathbf{u}_k}$ the *relative location* of $\mathbf{p}$ in $\mathbf{u}_k$ ($0\le\delta(\mathbf{p})\le\delta$) [1].

**S2** The obfuscated location $\tilde{\mathbf{p}}$ is required to have the same relative location with its true location $\mathbf{p}$, i.e., $\delta(\mathbf{p})=\delta(\tilde{\mathbf{p}})$ whichever interval is $\tilde{\mathbf{p}}$ in.

**S3** For any pair of true locations $\mathbf{p}_1$ and $\mathbf{p}_2$ that are in the same interval $\mathbf{u}_i$, the probabilities of their obfuscated location $\tilde{\mathbf{p}}_1$ and $\tilde{\mathbf{p}}_2$ in each interval $\mathbf{u}_l$ are the same, i.e.,

$$\Pr(\tilde{\mathbf{p}}_1\in\mathbf{u}_l|P=\mathbf{p}_1)=\Pr(\tilde{\mathbf{p}}_2\in\mathbf{u}_l|P=\mathbf{p}_2) \tag{14}$$

where $l=1,...,K$.

We note that Step II and Step III introduce additional constraints to VLP, and hence they shrink the feasible region of VLP [29], indicating that the optimal solution of D-VLP offers an *upper bound* of the minimum QoS loss in VLP.

**Proposition 3.1:** Suppose that a pair of true locations $\mathbf{p}_1$ and $\mathbf{p}_2$ are in the same interval $\mathbf{u}_i$. Then, in D-VLP:
A) Given any task location $\mathbf{q}$, $\mathbf{p}_1$ and $\mathbf{p}_2$ have the same estimation error of ShPD to $\mathbf{q}$, i.e.,

$$\mathbf{E}\left(\Delta d_{\mathcal{G}}\left(\mathbf{p}_1,\tilde{P}_1;\mathbf{q}\right)\right)=\mathbf{E}\left(\Delta d_{\mathcal{G}}\left(\mathbf{p}_2,\tilde{P}_2;\mathbf{q}\right)\right). \tag{15}$$

B) A location obfuscation strategy satisfies the $(\epsilon,r)$-Geo-I constraint for $\mathbf{p}_1$ if only if it satisfies the constraint for $\mathbf{p}_2$.

*Proof:* The detailed proof can be found in Appendix. ■

[1] Due to the variety of edge length, there exists some intervals with length smaller than $\delta$. But as $\delta$ is small enough, we won't discuss these intervals in the following part considering the tractability of our solution.

Proposition 3.1 indicates that we don't need to differentiate any pair of true locations within the same interval when calculating QoS loss or checking Geo-I. Accordingly, we can rewrite the objective function in VLP (Equation (12)) based on Proposition 3.1-A:

$$\mathbf{E}\left(\Delta d_{\mathcal{G}}\left(P,\tilde{P};Q\right)\right)=\sum_{i}\sum_{l}c_{i,l}z_{i,l} \tag{16}$$

where $z_{i,l}$ represents the probability that the obfuscated location $\tilde{\mathbf{p}}$ is in $\mathbf{u}_l$ given the true location $\mathbf{p}$ in $\mathbf{u}_i$ and

$$c_{i,l}=\int_{\mathbf{u}_i}\int_{\mathbf{u}_l}\int\Delta d_{\mathcal{G}}\left(\mathbf{p},\tilde{\mathbf{p}};\mathbf{q}\right)f_Q(\mathbf{q})f_P(\mathbf{p})\mathrm{d}\mathbf{q}\mathrm{d}\tilde{\mathbf{p}}\mathrm{d}\mathbf{p} \tag{17}$$

is a constant (note that $\Delta d_{\mathcal{G}}\left(\mathbf{p},\tilde{\mathbf{p}};\mathbf{q}\right)$, $f_Q(\mathbf{q})$, $f_P(\mathbf{p})$ are all known). Also, according to Proposition 3.1-B, the Geo-I constraint in VLP (Equation (13)) can be rewritten by

$$\begin{aligned}&z_{i,j}-e^{\epsilon d_{\mathcal{G}}^{\min}(\mathbf{u}_i^{\mathrm{e}},\mathbf{u}_l^{\mathrm{e}})}z_{l,j}\leq 0,\forall\mathbf{u}_i,\mathbf{u}_j,\mathbf{u}_l \\ &\text{s.t. } d_{\mathcal{G}}^{\min}(\mathbf{u}_i,\mathbf{u}_l)\leq r\end{aligned} \tag{18}$$

where $d_{\mathcal{G}}^{\min}(\mathbf{u}_i,\mathbf{u}_l)=d_{\mathcal{G}}^{\min}(\mathbf{u}_i^{\mathrm{s}},\mathbf{u}_l^{\mathrm{e}})$. Eventually, D-VLP can be written as a *linear programming* problem:

$$\min \quad \sum_{i}\sum_{l}c_{i,l}z_{i,j} \tag{19}$$

$$\text{s.t.} \quad \text{Equation (18) is satisfied.} \tag{20}$$

where the decision variables are $\mathbf{Z}=\{z_{i,j}\}_{K\times K}$. The D-VLP can be solved by many well-developed classic methods for linear programming such as the simplex methods [29].

### *B. Time Efficiency Improvement by Constraint Reduction*

According to the definition of $(\epsilon,r)$-Geo-I (Equation (18)), given any obfuscated interval $\mathbf{u}_j$ $(j=1,...,K)$, we need to set a constraint for each pair of $z_{i,j}$ and $z_{l,j}$ $(i,l=1,...,K)$, which generates $O(K^3)$ inequality constraints to D-VLP in total. Although linear programming is solvable by many existing approaches, it is crucial to reduce the huge number of constraints, which highly affects the time efficiency for solving the linear programming problem [29].

Fortunately, there are some features of partitioned intervals in road networks that can be exploited to reduce the number of inequality constraints in D-VLP. Along these features, we find that, to constraining all pairs of intervals partitioned in the road network, it is sufficient to apply Geo-I to pairs of intervals that are "adjacent" (Definition 3.1), achieving *constraint reduction*.

Before describing the constraint reduction, we first introduce Definition 3.1–3.2, Property 3.1–3.2, and Theorem 3.2.

**Definition 3.1:** (*Auxiliary graph*) We build a weighted directed auxiliary graph $\mathcal{G}'=(\mathcal{U}',\mathcal{E}')$, where the vertex set

$$\mathcal{U}'=\{\mathbf{u}_1',\mathbf{u}_2',...,\mathbf{u}_{|\mathcal{U}|}'\} \tag{21}$$

corresponds $\mathcal{U}$, and if any pair of $\mathbf{u}_i$ and $\mathbf{u}_l$ are adjacent in the road network and the worker can directly travel from $\mathbf{u}_i$ to $\mathbf{u}_l$, then we build a directed edge from $\mathbf{u}_i'$ to $\mathbf{u}_l'$ with weight $\delta$ in $\mathcal{G}'$ (Figure 5 gives an example).

The auxiliary graph $\mathcal{G}'$ is used to describe the relationship among intervals in $\mathcal{U}$, where the ShPD between any pair of vertices, say $\mathbf{u}_i'$ and $\mathbf{u}_l'$, equals to the ShPD between the corresponding intervals $\mathbf{u}_i$ and $\mathbf{u}_l$. Accordingly, checking Geo-I between $\mathbf{u}_i$ and $\mathbf{u}_l$ is equivalent to checking Geo-I for $\mathbf{u}_i'$ and $\mathbf{u}_l'$. With the auxiliary graph, we can directly apply some existing data structures (e.g., shortest path trees) to help implement the constraint reduction, where the details will be introduced in the algorithm later.

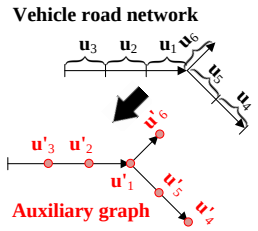


Figure 5. Auxiliary graph.

**Definition 3.2:** We use $\mathbf{u}_i' \overset{G(\epsilon,r)}{\simeq} \mathbf{u}_l'$ to denote that Geo-I is satisfied in the direction from $\mathbf{u}_i'$ to $\mathbf{u}_l'$. More precisely, $\mathbf{u}_i' \overset{G(\epsilon,r)}{\simeq} \mathbf{u}_l'$ if $z_{i,j}-e^{\epsilon d_{\mathcal{G}}(\mathbf{u}_i^{\mathrm{e}},\mathbf{u}_l^{\mathrm{e}})}z_{l,j}\leq 0,\ \forall\mathbf{u}_j$.

According to Definition 2.1 and Definition 3.2, it is trivial to obtain Property 3.1–3.2:

**Property 3.1:** $\mathbf{u}_i'$ and $\mathbf{u}_l'$ satisfies $(\epsilon,r)$-Geo-I constraint if only if $\mathbf{u}_i' \overset{G(\epsilon,r)}{\simeq} \mathbf{u}_l'$ and $\mathbf{u}_l' \overset{G(\epsilon,r)}{\simeq} \mathbf{u}_i'$.

**Property 3.2:** (*Transitivity*) Given a shortest path from $\mathbf{u}_i'$ to $\mathbf{u}_k'$ that is composed of two edges in $\mathcal{G}'$, say $(\mathbf{u}_i',\mathbf{u}_l')\rightarrow(\mathbf{u}_l',\mathbf{u}_k')$, then:

$$\mathbf{u}_i' \overset{G(\epsilon,r)}{\simeq} \mathbf{u}_l' \text{ and } \mathbf{u}_l' \overset{G(\epsilon,r)}{\simeq} \mathbf{u}_k' \Rightarrow \mathbf{u}_i' \overset{G(\epsilon,r)}{\simeq} \mathbf{u}_k'. \tag{22}$$

*Proof:* $\forall\mathbf{u}_j'$, if $\mathbf{u}_i' \overset{G(\epsilon,r)}{\simeq} \mathbf{u}_l'$ and $\mathbf{u}_l' \overset{G(\epsilon,r)}{\simeq} \mathbf{u}_k'$, we have

$$\mathbf{u}_i' \overset{G(\epsilon,r)}{\simeq} \mathbf{u}_l' \quad\Rightarrow\quad z_{i,j}\leq e^{\epsilon d_{\mathcal{G}}(\mathbf{u}_i^{\mathrm{e}},\mathbf{u}_l^{\mathrm{e}})}z_{l,j} \tag{23}$$

$$\mathbf{u}_l' \overset{G(\epsilon,r)}{\simeq} \mathbf{u}_k' \quad\Rightarrow\quad z_{l,j}\leq e^{\epsilon d_{\mathcal{G}}(\mathbf{u}_l^{\mathrm{e}},\mathbf{u}_k^{\mathrm{e}})}z_{k,j}. \tag{24}$$

from which we can obtain that

$$\begin{aligned}z_{i,j} &\leq e^{\epsilon d_{\mathcal{G}}(\mathbf{u}_i^{\mathrm{e}},\mathbf{u}_l^{\mathrm{e}})}z_{l,j}\leq e^{\epsilon(d_{\mathcal{G}}(\mathbf{u}_i^{\mathrm{e}},\mathbf{u}_l^{\mathrm{e}})+d_{\mathcal{G}}(\mathbf{u}_l^{\mathrm{e}},\mathbf{u}_k^{\mathrm{e}}))}z_{k,j} \\ &= e^{\epsilon(d_{\mathcal{G}}(\mathbf{u}_i^{\mathrm{e}},\mathbf{u}_k^{\mathrm{e}}))}z_{k,j}\end{aligned} \tag{25}$$

indicating that $\mathbf{u}_i' \overset{G(\epsilon,r)}{\simeq} \mathbf{u}_k'$. ■

In Theorem 3.2, we generalize *transitivity* by induction:

**Theorem 3.2:** Given any pair of vertices $\mathbf{u}_1'$ and $\mathbf{u}_K'$, suppose that a *shortest path* from $\mathbf{u}_1'$ to $\mathbf{u}_K'$ is composed of $K-1$ edges in $\mathcal{G}'$: $(\mathbf{u}_1',\mathbf{u}_2')\rightarrow...\rightarrow\left(\mathbf{u}_{K-1}',\mathbf{u}_K'\right)$, then

$$\mathbf{u}_k' \overset{G(\epsilon,r)}{\simeq} \mathbf{u}_{k+1}',\ (k=1,...,K-1)\Rightarrow \mathbf{u}_1' \overset{G(\epsilon,r)}{\simeq} \mathbf{u}_K' \tag{26}$$

*Proof: Base case*: When $K=3$, Theorem 3.2 is exactly the same as *transitivity* that we have obtained in Property 3.2.

*Step case*: Assuming that the statements holds for $K=n$, then when $K=n+1$, given $\mathbf{u}_k' \overset{G(\epsilon,r)}{\simeq} \mathbf{u}_{k+1}'$, $(k=1,...,n)$, we have $\mathbf{u}_k' \overset{G(\epsilon,r)}{\simeq} \mathbf{u}_{k+1}'$, $(k=1,...,n-1)\Rightarrow \mathbf{u}_1' \overset{G(\epsilon,r)}{\simeq} \mathbf{u}_n'$,

then $\mathbf{u}'_1 \overset{G(\epsilon,r)}{\simeq} \mathbf{u}'_n$ and $\mathbf{u}'_n \overset{G(\epsilon,r)}{\simeq} \mathbf{u}'_{n+1} \Rightarrow \mathbf{u}'_1 \overset{G(\epsilon,r)}{\simeq} \mathbf{u}'_{n+1}$, indicating that the statement holds for $K = n+1$. ■

According to Theorem 3.2, to provide a sufficient condition for the Geo-I constraint for any pair of vertices $\mathbf{u}'_i$ and $\mathbf{u}'_l$ in $\mathcal{U}$', we can 1) first find the shortest path between $\mathbf{u}'_i$ and $\mathbf{u}'_l$ in both directions (from $\mathbf{u}'_i$ to $\mathbf{u}'_l$ and from $\mathbf{u}'_l$ to $\mathbf{u}'_i$), and then 2) select the shortest path between the two paths, say $\mathcal{P}$, and set the Geo-I constraint $\mathbf{u}'_k \overset{G(\epsilon,r)}{\simeq} \mathbf{u}'_{k+1}$ for each pair of adjacent vertices $\mathbf{u}'_k$ and $\mathbf{u}'_{k+1}$ in $\mathcal{P}$. We repeat such process for all pairs of vertices in $\mathcal{G}'$.

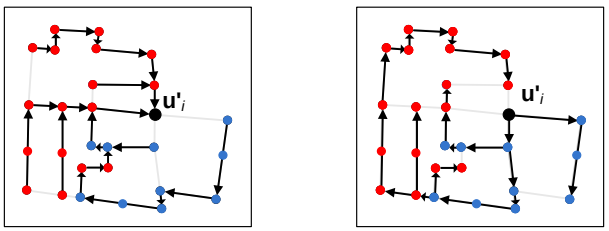
In the above process of constraint reduction, any constraint constructed by adjacent vertices, say $\mathbf{u}'_k \overset{G(\epsilon,r)}{\simeq} \mathbf{u}'_{k+1}$, won't shrink the feasible region of the D-VLP, since $\mathbf{u}'_k$ and $\mathbf{u}'_{k+1}$ themselves also need to satisfy the Geo-I constraint (according to Property 3.1). Hence, *the optimality of D-VLP will not be lost by the constraint reduction.*

**Algorithm 1:** Pseudo-code of constraint reduction.

**input** : $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
**output** : $\mathbf{U}_{\text{con}} = \{u_{i,j}\}_{|\mathcal{U}|\times|\mathcal{U}|}$
1 Initialize $\mathbf{U}_{\text{con}}$ by $\mathbf{0}$;
2 Initialize the sets $\mathcal{U}'_{\text{In},1}, ..., \mathcal{U}'_{\text{In},|\mathcal{U}|}$, $\mathcal{U}'_{\text{Out},1}, ..., \mathcal{U}'_{\text{Out},|\mathcal{U}|}$ by empty;
3 **for** *each* $\mathbf{u}_i \in \mathcal{V}$ **do**
4  Build both SPT-In($i$) and SPT-Out($i$);
5  **for** *each* $\mathbf{u}_j \in \mathcal{V}\backslash\mathbf{u}_i$ **do**
6  **if** ShPD($\mathbf{u}_i$, $\mathbf{u}_j$) $\leq$ ShPD($\mathbf{u}_j$, $\mathbf{u}_i$) **then**
7  add $\mathbf{u}'_j$ to $\mathcal{U}'_{\text{Out},i}$;
8  **otherwise do**
9  add $\mathbf{u}'_j$ to $\mathcal{U}'_{\text{In},i}$;
10  **for** *each* $\mathbf{u}'_j \in \mathcal{U}'_{\text{Out},i}$ **do**
11  Traverse the edges in the path from $\mathbf{u}'_j$ to $\mathbf{u}'_i$ and let $u_{l,k} = 1$ if $(\mathbf{u}'_l, \mathbf{u}'_k)$ is an edge in the path;
12  **for** *each* $\mathbf{u}'_j \in \mathcal{U}'_{\text{I},i}$ **do**
13  Traverse the edges in the path from $\mathbf{u}'_j$ to $\mathbf{u}'_i$ and let $u_{l,k} = 1$ if $(\mathbf{u}'_l, \mathbf{u}'_k)$ is an edge in the path;
14 **return** $\mathbf{U}_{\text{con}}$;

Since each pair of adjacent vertices in $\mathcal{P}$ must be the two endpoints of an edge in $\mathcal{G}'$, the number of possible adjacent vertices in all shortest paths cannot exceed $M$ ($M$ denotes the number of edges in the auxiliary graph $\mathcal{G}'$, i.e., $M = |\mathcal{E}'|$). For each obfuscated vertex $\mathbf{u}'_j$ ($j = 1, ..., K$) (i.e., the obfuscated location is in $\mathbf{u}_j$), instead of building constraint for each pair of vertices in $\mathcal{U}'$, we only need to build up to $M$ constraints for the pairs that are adjacent. Hence, the total number of constraints to be added is $O(KM)$. According to the trace of some real world road networks, $M$ is close to $K$, where the detailed observation will be introduced in Section IV (Figure 11). Accordingly, the number of constraints in D-VLP can be reduced from $O(K^3)$ to approximately $O(K^2)$.

Algorithm 1 gives the pseudo code of our constraint reduction method: We use an indicator matrix $\mathbf{U}_{\text{con}} = \{u_{i,j}\}_{K\times K}$ to represent whether a constraint for $\mathbf{u}'_i$ and $\mathbf{u}'_j$ is added: if the pair $\{\mathbf{u}'_i, \mathbf{u}'_j\}$ needs a constraint, $u_{i,j} = 1$; otherwise, $u_{i,j} = 0$. To find the shortest path between all pairs of vertices in $\mathcal{G}'$, we build the *shortest path trees (SPTs)* [33] rooted at each vertex $\mathbf{u}'_i$ ($i = 1, ..., |\mathcal{U}|$), respectively.




(a) SPT-Out($i$) (b) SPT-In($i$)

Figure 6. An example of the two types of SPTs for the vertex $\mathbf{u}'_i$ (in both figures, $\mathcal{U}'_{\text{In},i}$ and $\mathcal{U}'_{\text{Out},i}$ are respectively marked by red color and blue color).

We note that the shortest path between any pair of vertices can be in two directions and we only need to check Geo-I for the shorter one. Here, for each $\mathbf{u}'_i$, we build two SPTs: SPT-Out($i$) and SPT-In($i$) (as shown in Figure 6(a)(b)), in which all the paths take $\mathbf{u}'_i$ as the source and the destination, respectively (line 4). After building a SPT (can be either SPT-Out($i$) or SPT-In($i$)), it is unnecessary to find the path for each vertex to (or from) $\mathbf{u}'_i$ in the tree, since the vertex may have a shorter path with $\mathbf{u}'_i$ in the other tree. Hence, before finding the paths, we categorize all the vertices in $\mathcal{U}'\backslash\mathbf{u}'_i$ into two subsets $\mathcal{U}'_{\text{In},i}$ and $\mathcal{U}'_{\text{Out},i}$ based on whether each vertex in $\mathcal{U}'\backslash\mathbf{u}'_i$ has a shorter path to (or from) $\mathbf{u}'_i$ in SPT-In($i$) than in SPT-Out($i$) (line 5-9). After the vertex categorization, in SPT-In($i$), all the paths from the vertices in $\mathcal{U}'_{\text{In},i}$ to $\mathbf{u}'_i$ are collected; and in SPT-Out($i$), all the paths from $\mathbf{u}'_i$ to the vertices in $\mathcal{U}'_{\text{Out},i}$ are collected. Finally, for each pair of adjacent vertices in the collected paths, we add the corresponding constraint to D-VLP (line 10–13).

**Time complexity analysis of the constraint reduction**. The constraint reduction mainly includes SPT building (line 4), vertex categorization (line 5–9), and constraint addition (line 10–13) for each $\mathbf{u}'_i$ ($i = 1, ..., |\mathcal{U}|$). We adopt a well-developed method Dijkstra [33] to build the SPTs, of which the time complexity is $O(M + K\log K)$. Vertex categorization requires to compare the length of each vertex's two paths with $\mathbf{u}'_i$, taking up to $K$ comparisons. Constraint addition requires to traverse all the edges in the two SPTs, both of which have up to $K$ edges. Eventually, the time complexity of the constraint reduction method can be calculated by

$$\begin{aligned} T_{\text{cr}} &= O(K) \times (\underbrace{O(M + K\log K)}_{\text{line 4}} + \underbrace{O(K)}_{\text{line 5-9}} + \underbrace{O(K)}_{\text{line 10–13}}) \\ &= O\left(MK + K^2\log K + K^2\right). \end{aligned} \tag{27}$$

### C. Lower Bound Derivation

For theoretical interests, we also derive a lower bound of the minimum QoS loss in VLP via *problem relaxation* [29]. By comparing this lower bound with the QoS loss obtained

by our approach, we can check how close the solution can achieve the optimal.

To derive a lower bound via problem relaxation, we need to create a relaxed version of VLP that can be solved with low time complexity, and meanwhile, the relaxed problem

1) has the objective function upper bounded by the QoS loss defined in VLP *(Condition A)*,
2) has its feasible region as a super-set of the VLP's feasible region *(Condition B)*.

As such, the optimal solution of this relaxed VLP will be a lower bound of the minimum QoS loss in VLP [29].

Following this idea, we define a relaxed VLP (denoted by R-VLP) as follows:

$$\min \quad \sum_i \sum_j c'_{i,j} w_{i,j} \tag{28}$$

$$\text{s.t.} \quad \Omega(\mathbf{W}) \quad (\mathbf{W} \text{ is a matrix } \{w_{i,j}\}_{M\times M}) \tag{29}$$

where $w_{i,j} = \int_{e(\mathbf{p})} \int_{e(\tilde{\mathbf{p}})} f_{\tilde{P}}(\tilde{\mathbf{p}}|P=\mathbf{p}) f_P(\mathbf{p}) \mathrm{d}\tilde{\mathbf{p}}\mathrm{d}\mathbf{p}$ is the *decision variable*, $c'_{i,j} = \int \Delta \underline{d}_{\mathcal{G}}(\mathbf{p}, \tilde{\mathbf{p}}; \mathbf{q}) f_Q(\mathbf{q}) \mathrm{d}\mathbf{q}$ (given $e(\mathbf{p})$ and $e(\tilde{\mathbf{p}})$ are $i$th and $j$th edges respectively) is a constant ($i, j = 1, ..., M$) (as both $\Delta \underline{d}_{\mathcal{G}}(\mathbf{p}, \tilde{\mathbf{p}}; \mathbf{q})$ and $f_Q(\mathbf{q})$ are known), $\Omega(\mathbf{W})$ is a polyhedron defined by

$$\Omega(\mathbf{W}) = \left\{ \mathbf{W} \;\middle|\; \begin{array}{l} \frac{1}{\int_{e_i} f_P(\mathbf{p})\mathrm{d}\mathbf{p}} w_{i,j} \le \frac{e^{\epsilon d_{\mathcal{G}}(e_i,e_l)}}{\int_{e_l} f_P(\mathbf{p})\mathrm{d}\mathbf{p}} w_{l,j}, \\ \forall j \text{ and } d_{\mathcal{G}}(e_i, e_l) \le r. \\ \sum_i \sum_j w_{i,j} = 1 \end{array} \right\}, \tag{30}$$

$d_{\mathcal{G}}(e_i, e_l)$ denotes the longest ShPD between the points in $e_i$ and the points in $e_l$, and $\Delta \underline{d}_{\mathcal{G}}(\mathbf{p}, \tilde{\mathbf{p}}; \mathbf{q})$ is defined by

$$C(1,1): \ \Delta \underline{d}_{\mathcal{G}}(\mathbf{p}, \tilde{\mathbf{p}}; \mathbf{q}) \tag{31}$$

$$= \begin{cases} \Delta d_{\mathcal{G}}(v^{\mathrm{e}}_{e(\mathbf{p})}, v^{\mathrm{e}}_{e(\tilde{\mathbf{p}})}, v^{\mathrm{s}}_{e(\mathbf{q})}) - l_{e_{\tilde{\mathbf{p}}}} \\ \text{if } d_{\mathcal{G}}(v^{\mathrm{e}}_{e(\mathbf{p})}, v^{\mathrm{s}}_{e(\mathbf{q})}) \ge d_{\mathcal{G}}(v^{\mathrm{e}}_{e(\tilde{\mathbf{p}})}, v^{\mathrm{s}}_{e(\mathbf{q})}) + l_{e_{\tilde{\mathbf{p}}}} \\ \Delta d_{\mathcal{G}}(v^{\mathrm{e}}_{e(\tilde{\mathbf{p}})}, v^{\mathrm{e}}_{e(\mathbf{p})}, v^{\mathrm{s}}_{e(\mathbf{q})}) - l_{e(\mathbf{p})} \\ \text{if } d_{\mathcal{G}}(v^{\mathrm{e}}_{e(\tilde{\mathbf{p}})}, v^{\mathrm{s}}_{e(\mathbf{q})}) \ge d_{\mathcal{G}}(v^{\mathrm{e}}_{e(\mathbf{p})}, v^{\mathrm{s}}_{e(\mathbf{q})}) + l_{e(\mathbf{p})} \\ 0 \quad \text{otherwise} \end{cases}$$

$$C(1,2): \ \Delta \underline{d}_{\mathcal{G}}(\mathbf{p}, \tilde{\mathbf{p}}; \mathbf{q}) = d_{\mathcal{G}}(v^{\mathrm{e}}_{e(\mathbf{p})}, v^{\mathrm{s}}_{e(\mathbf{q})}) \tag{32}$$

$$C(2,1): \ \Delta \underline{d}_{\mathcal{G}}(\mathbf{p}, \tilde{\mathbf{p}}; \mathbf{q}) = d_{\mathcal{G}}(v^{\mathrm{e}}_{e(\tilde{\mathbf{p}})}, v^{\mathrm{s}}_{e(\mathbf{q})}) \tag{33}$$

$$C(2,2): \ \Delta \underline{d}_{\mathcal{G}}(\mathbf{p}, \tilde{\mathbf{p}}; \mathbf{q}) = 0, \tag{34}$$

(different cases $C(i,j)$ are defined in Section II-B).

**Proposition 3.3:** The optimal solution of R-VLP problem offers a lower bound of the minimum QoS loss in VLP.

*Proof:* The basic idea is to prove that Condition A&B are satisfied in R-VLP. The detailed proof can be found in Appendix. ■

R-VLP can be solved directly by applying linear programming approaches such as the simplex methods [29] since its number of constraints $O(M^2)$ is acceptable. We compare the lower bound derived from this problem with our solution in the performance evaluation part based a real world dataset [30] (in Figure 8(a)(b)).

## IV. Performance Evaluation

In this section, we turn our attention to practical applications of our location obfuscation approach. The two main metrics we will measure include:

1) *The estimation error of ShPD*, which is defined in Equation (11). We use this metric to reflect the QoS loss of location obfuscation strategies.

2) *The best guess of the adversary given the reported*, or *AdvError* for short [23]. Here we assume the adversary use the *optimal inference attack* [26]. We adopt this metric to reflect the *privacy level* that our approach can achieve, where higher AdvError indicates higher privacy level.

**Dataset**. We test the performance of our approach with a real world dataset, and use the dataset provided by [30], which records the trajectories of taxi cabs in Rome, as taxi services are considered as a type of VSC [30]. The dataset contains GPS coordinates of approximately 290 taxis collected over 30 days. Figure 7 depicts the heat map of all taxi cabs' recorded location, from which we can observe that taxi cabs' locations are not evenly distributed over the city, e.g., on average taxi cabs are more likely located in downtown than in the suburbs. Note that cabs may have different number of records (including both location and time stamp) in the trace. Here, we select the 120 cabs with the highest number of records in the trace and estimate each cab's prior probability distribution $f_P(\mathbf{p})$ based on its own records. Then, we conduct a simulation for each single cab, where we randomly pick up a location on the road network based on the vehicle's $f_P(\mathbf{p})$, and find the obfuscated location with our approach. In addition, we assume that the task's (customer's) location has the same probability distribution with the location of all cabs [34].

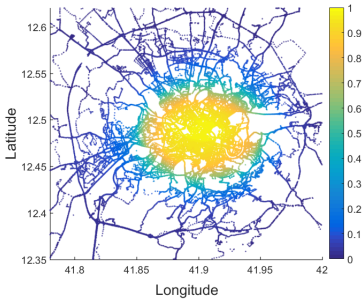
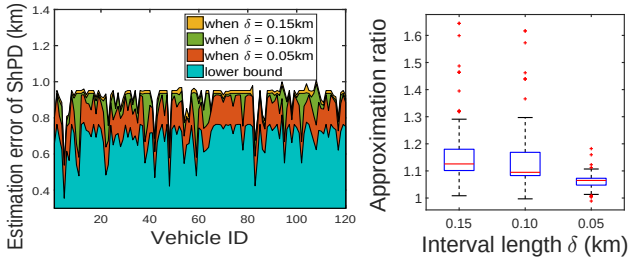



Figure 7. Cabs' location distribution.

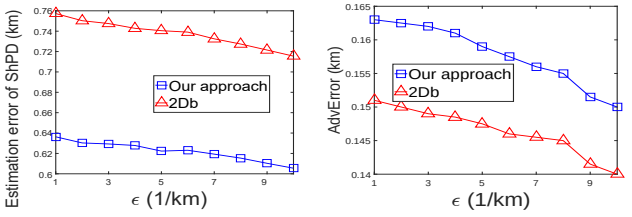
### A. Comparison with the lower bound

As the minimum QoS loss in VLP is within the gap between its lower bound (derived in Section III-C) and the QoS loss of our approach, it is interesting to check how close our approach can achieve the optimal by comparing the QoS achieved by our approach with the lower bound. Figure 8(a) compares the QoS loss of our approach and the lower bound for 120 cabs, where we set the interval size $\delta$ in the D-VLP by 0.05km, 0.1km, and 0.15km respectively. From the figure, we can observe that the denser we partition the road network in the approximation algorithm, the closer our solution can achieve the lower bound. This observation is confirmed in




(a) Comparison with the lower bound (b) Approximation ratio

Figure 8. Performance of the cloaking strategy with different $\delta$.




(a) QoS loss (b) Privacy

Figure 9. Comparison of our approach with a 2D based method.

Figure 8(b), where we calculate the approximation ratio of our approach by taking the ratio between our approach's QoS loss and the lower bound, and give the box plot of the approximation ratio for 120 cabs with different $\delta$. The figure shows that, as $\delta$ decreases ($\delta = 0.15km, 0.1km, 0.05km$), the approximation ratio decreases (which are 1.17, 1.11, and 1.06, respectively). Note that when the approximation ratio = 1, the solution achieves the optimal.

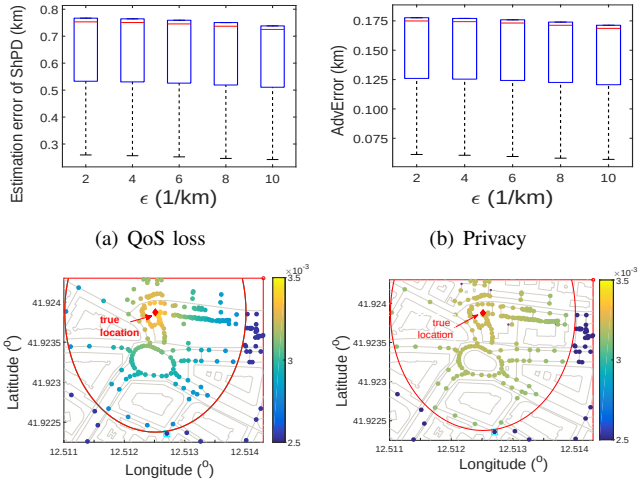
### B. Comparison with 2D-plane based methods

We also compare our approach with the existing 2D based location obfuscation methods. Here, we pick a state of the art mechanism introduced in [23] as baseline. Note that [23] is also a global optimization framework: given the privacy constraint, the object is to minimize the QoS loss. Different from our approach, this 2D-based method (or simply *2Db*) assume locations on a 2D plane and both QoS loss and privacy are measured by the Euclidean distance[2].

Figure 9(a)(b) compare the average QoS loss and AdvError of 120 cabs by using our approach and 2Db, respectively. Not surprisingly, our approach outperforms 2Db in both metrics since 2Db neglects the structure feature of road network. For example, a pair of locations with shorter Euclidean distance may take longer path distance in the road network.

### C. Performance given different threshold values for $\epsilon$

Besides testing our approach's effectiveness, we check how the parameter $\epsilon$ in Geo-I will impact the QoS loss and privacy of our method in Figure 10(a)(b), in which $\epsilon$ is changed from 2/km to 10/km. From the figures, we observe that larger $\epsilon$ generates lower QoS loss and lower AdvError. According to the definition of $(\epsilon, r)$-Geo-I (Definition 2.1), with higher $\epsilon$ (e.g., $\epsilon = 10/km$), the obfuscated location probability can be less evenly distributed over the road network. Hence, the obfuscated location around the true location will have higher probability to be selected, as shown in the heat map in Figure 10(c), leading to a lower QoS loss and a lower AdvError. In contrast, when $\epsilon$ is lower (e.g., $\epsilon = 2/km$), the obfuscated location probability is required to be more evenly distributed, as shown in the heat map in Figure 10(c). Then, obfuscated locations with relatively higher ShPD from (to) the true location will have higher probability to be selected, which causes higher QoS loss and AdvError. According to Figure 10(a)(b), we also find that it is difficult to increase both QoS and privacy at the same time. As expected, it is very important to take a trade-off between the two objectives based on workers' preference.




(a) QoS loss (b) Privacy

(c) Obfuscation location probability distribution when $\epsilon = 10$/km (d) Obfuscation location probability distribution when $\epsilon = 2$/km

Figure 10. Performance of the cloaking strategy with different $\epsilon$.

### D. Computation time

Finally, we test how the constraint reduction (CR) method introduced in Section III-B can improve the time efficiency of our approach in Figure 11. From the figure, we find that CR largely reduces the computation time (on average by 87.9%). In addition, when $\delta = 0.15km, 0.1km, 0.05km$, the number of edges in the auxiliary graph are only 56.7%, 28.4%, and 18.9% higher than the number of vertices in the graph, indicating that CR re-

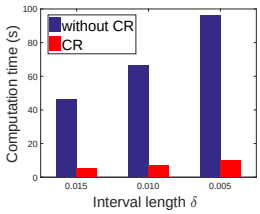



Figure 11. Computation time with and without CR.

[2]Note that 2Db may choose an obfuscated location that is not included in any edge in the network, so given an obfuscated location calculated by 2Db, we assume that the adversary will take the location in the road network that has the shortest Euclidean distance to this obfuscated location as the "reported location" from the worker

duces the number of constraints in D-VLP from cubic to approximately quadratic with the respect to the number of intervals partitioned in D-VLP.

## V. Related Work

During the last decade, a variety of location privacy protection approaches have been developed. Many of these methods allow users to hide his identity from the server, such as *k-anonymity* (i.e., a user's location cannot be distinguished with the other $k-1$ users) [12]–[14], *cloaking* (i.e., the accurate location is hidden in a obfuscated region) [15], [16]. Some other works let users use pseudonyms to interact with the system, where the users can change their pseudonyms without being traced by the system [17]. These approaches cannot be applied to VSC, since in VSC workers' identity has to be known by the server for task distribution. Although *obfuscation* has been also widely used for protecting location privacy [18]–[22], [26]. It introduces errors to location-based services, and hence one key problem is how to establish a trade-off between QoS and privacy. For example, the strategies introduced in [23]–[25] follow a global optimization framework, which QoS (or privacy) constraints are satisfied.

*Differential privacy* [35]–[37] has also been applied to address location privacy issues, though many of these works are used to protect aggregate location information [38]–[40], which is much different from the problem we discuss in this paper. As apposed to requiring low sensitivity of aggregate output to a single individual change, the notion of Geo-I we adopt in this paper sets constraints such that any small change of location will not have a significant effect on adversary's observation. Following this notion, many Geo-I based location obfuscation strategies have been proposed [23]–[26]. Recently, some researchers have started working on location privacy issues of some specific location based services (LBS), such as mobile spatial crowdsourcing (SC) [8]–[10]. Similar to our work, most of these methods target maximizing the reacheability from workers to spatial tasks without compromising workers' location privacy. However, all these works assume both workers' and tasks' location on a 2D plane. As we have demonstrated that 2D based strategy cannot effectively achieve high QoS and high privacy in a vehicle road network, these existing approaches cannot be applied to VSC.

To date, the work closest to ours is [23]. [23] proposes an optimal location obfuscation mechanism with regard to Geo-I based on LP. However, their approach still assumes users' locations on a 2D plane and hence cannot be applied in VSC. On the other hand, although [23] also proposes an approximation technique to reduce the number of constraints in LP, their approach may not guarantee optimal solutions as it shrinks the LP's feasible region. Instead, by exploiting some unique features of Geo-I on road networks, our constraint reduction approach can significantly reduce the computation time without losing the optimality of the original problem.

## VI. Conclusions

In this paper, we designed a location obfuscation strategy to minimize the QoS loss of task distribution without compromising workers' location privacy, as defined by Geo Indisitiguishability constraints. Through discretization, we approximated our obfuscation problem as a linear programming problem that can be solved and further reduce its complexity by constraint reduction. Finally, our experimental results demonstrate that our approach can well approximate the optimal QoS, and also outperforms state-of-the-art location obfuscation strategy in terms of both QoS and privacy.

We see a number of promising directions for this research work. For exmaple, we plan to further investigate VSC privacy frameworks in heterogeneous settings, in which users may have different QoS preferences over different regions in the road network, e.g., some workers may tolerate less QoS loss in downtown than in suburban areas. Another direction we can explore is to further increase the scalability of our current approach. Although the time efficiency of our current approach has been improved significantly, it still brings challenges when applying our approach to large scale systems that are composed of a huge number of vehicle workers, especially when workers' interaction needs to be considered. One potential solution is to implement our approach in a decentralized way by resorting to optimization decomposition techniques.

## Appendix

### A. Proof of Proposition 3.1

*Proof:* A) Step III ensures $\mathbf{p}_1$ and $\mathbf{p}_2$ have the same obfuscated location probability distribution over different intervals. $\forall \mathbf{u}_k$, suppose there exist obfuscated locations $\tilde{\mathbf{p}}_1$ and $\tilde{\mathbf{p}}_2$ (for $\mathbf{p}_1$ and $\mathbf{p}_2$ respectively) in $\mathbf{u}_k$,

$$\Delta d_{\mathcal{G}}\left(\mathbf{p}_1, \tilde{\mathbf{p}}_1; \mathbf{q}\right) = |d_{\mathcal{G}}(\mathbf{u}_k^{\mathrm{s}}, \mathbf{q}) + \delta(\mathbf{p}_1) - d_{\mathcal{G}}(\mathbf{u}_i^{\mathrm{s}}, \mathbf{q}) - \delta(\tilde{\mathbf{p}}_1)|$$

$$\Delta d_{\mathcal{G}}\left(\mathbf{p}_2, \tilde{\mathbf{p}}_2; \mathbf{q}\right) = |d_{\mathcal{G}}(\mathbf{u}_k^{\mathrm{s}}, \mathbf{q}) + \delta(\mathbf{p}_2) - d_{\mathcal{G}}(\mathbf{u}_i^{\mathrm{s}}, \mathbf{q}) - \delta(\tilde{\mathbf{p}}_2)|$$

Then, according to Step II, $\delta(\mathbf{p}_1) = \delta(\tilde{\mathbf{p}}_1)$, $\delta(\mathbf{p}_2) = \delta(\tilde{\mathbf{p}}_2)$, implying that $\Delta d_{\mathcal{G}}\left(\mathbf{p}_1, \tilde{\mathbf{p}}_1; \mathbf{q}\right) = \Delta d_{\mathcal{G}}\left(\mathbf{p}_2, \tilde{\mathbf{p}}_2; \mathbf{q}\right)$. Finally, we obtain that $\mathbf{E}\left(\Delta d_{\mathcal{G}}\left(\mathbf{p}_1, \tilde{P}_1; \mathbf{q}\right)\right) = \mathbf{E}\left(\Delta d_{\mathcal{G}}\left(\mathbf{p}_2, \tilde{P}_2; \mathbf{q}\right)\right)$.

B) $\forall \mathbf{u}_k$, suppose there exists $\tilde{\mathbf{p}}_1$ and $\tilde{\mathbf{p}}_2$ in $\mathbf{u}_k$. For any other true locations $\mathbf{p}_1'$ and $\mathbf{p}_2'$ such that $\delta(\mathbf{p}_1') = \delta(\mathbf{p}_1)$, $\delta(\mathbf{p}_2') = \delta(\mathbf{p}_2)$, and $\mathbf{p}_1'$ and $\mathbf{p}_2'$ are in the same interval, we have the following relationships:

$$\Pr\left(\tilde{P} = \tilde{\mathbf{p}}_1 | P = \mathbf{p}_1\right) = \Pr\left(\tilde{P} = \tilde{\mathbf{p}}_2 | P = \mathbf{p}_2\right) \tag{35}$$

$$\Pr\left(\tilde{P} = \tilde{\mathbf{p}}_1 | P = \mathbf{p}_1'\right) = \Pr\left(\tilde{P} = \tilde{\mathbf{p}}_2 | P = \mathbf{p}_2'\right), \tag{36}$$

from which we can derive that

$$\Pr\left(\tilde{P} = \tilde{\mathbf{p}}_1 | P = \mathbf{p}_1\right) \leq e^{\epsilon d_{\mathcal{G}}^{\min}(\mathbf{p}_i, \mathbf{p}_l)} \Pr\left(\tilde{P} = \tilde{\mathbf{p}}_1 | P = \mathbf{p}_1'\right)$$

if only if

$$\Pr\left(\tilde{P} = \tilde{\mathbf{p}}_1 | P = \mathbf{p}_1\right) \leq e^{\epsilon d_{\mathcal{G}}^{\min}(\mathbf{p}_i, \mathbf{p}_l)} \Pr\left(\tilde{P} = \tilde{\mathbf{p}}_1 | P = \mathbf{p}_1'\right).$$

■

### B. Proof of Proposition 3.3

*Proof:* We prove Proposition 3.3 by showing that R-VLP satisfies Condition A and Condition B respectively.

*1) Condition A:* We first prove that $\Delta \underline{d}_{\mathcal{G}}(\mathbf{p}, \tilde{\mathbf{p}}; \mathbf{q})$ defined by Equations (31)-(34) is upper bounded by $\Delta d_{\mathcal{G}}(\mathbf{p}, \tilde{\mathbf{p}}; \mathbf{q})$ $\forall \mathbf{p}, \tilde{\mathbf{p}}, \mathbf{q}$ in all different cases: $C(1,1)$, $C(1,2)$, $(2,1)$, and $(2,2)$ (the proof of the inequality in $C(2,2)$ is trivial since $\Delta d_{\mathcal{G}}(\mathbf{p}, \tilde{\mathbf{p}}; \mathbf{q})$ is non-negative):

$C(1,1)$: When $d_{\mathcal{G}}(v^{\mathrm{e}}_{e(\mathbf{p})}, v^{\mathrm{s}}_{e(\mathbf{q})}) \geq d_{\mathcal{G}}(v^{\mathrm{e}}_{e(\tilde{\mathbf{p}})}, v^{\mathrm{s}}_{e(\mathbf{q})}) + l_{e_{\tilde{\mathbf{p}}}}$, since $l_{e_{\tilde{\mathbf{p}}}} \geq x_{\tilde{\mathbf{p}}} - x_{\mathbf{p}}$

$$\begin{aligned} & \Delta \underline{d}_{\mathcal{G}}(\mathbf{p}, \tilde{\mathbf{p}}; \mathbf{q}) \\ \leq\ & d_{\mathcal{G}}(v^{\mathrm{e}}_{e(\mathbf{p})}, v^{\mathrm{s}}_{e(\mathbf{q})}) - d_{\mathcal{G}}(v^{\mathrm{e}}_{e(\tilde{\mathbf{p}})}, v^{\mathrm{s}}_{e(\mathbf{q})}) - (x_{\tilde{\mathbf{p}}} - x_{\mathbf{p}}) \\ \leq\ & \left| d_{\mathcal{G}}(v^{\mathrm{e}}_{e(\mathbf{p})}, v^{\mathrm{s}}_{e(\mathbf{q})}) - d_{\mathcal{G}}(v^{\mathrm{e}}_{e(\tilde{\mathbf{p}})}, v^{\mathrm{s}}_{e(\mathbf{q})}) + x_{\mathbf{p}} - x_{\tilde{\mathbf{p}}} \right| \\ =\ & \Delta d_{\mathcal{G}}(\mathbf{p}, \tilde{\mathbf{p}}; \mathbf{q}) \ \text{(by triangle inequality)} \end{aligned}$$

Similar proof can be applied when

$$d_{\mathcal{G}}(v^{\mathrm{e}}_{e(\mathbf{p})}, v^{\mathrm{s}}_{e(\mathbf{q})}) \geq d_{\mathcal{G}}(v^{\mathrm{e}}_{e(\tilde{\mathbf{p}})}, v^{\mathrm{s}}_{e(\mathbf{q})}) + l_{e_{\tilde{\mathbf{p}}}}. \tag{37}$$

$C(1,2)$: Since $l_{e(\mathbf{q})} - x_{\tilde{\mathbf{p}}} \geq 0$ and $x_{\mathbf{p}} \geq 0$,

$$\Delta \underline{d}_{\mathcal{G}}(\mathbf{p}, \tilde{\mathbf{p}}; \mathbf{q}) \leq \left| d_{\mathcal{G}}(v^{\mathrm{e}}_{e(\mathbf{p})}, v^{\mathrm{s}}_{e(\mathbf{q})}) + x_{\mathbf{p}} \right| \leq \Delta d_{\mathcal{G}}(\mathbf{p}, \tilde{\mathbf{p}}; \mathbf{q}).$$

$C(2,1)$: Since $l_{e(\mathbf{q})} - x_{\mathbf{p}} \geq 0$ and $x_{\tilde{\mathbf{p}}} \geq 0$,

$$\Delta \underline{d}_{\mathcal{G}}(\mathbf{p}, \tilde{\mathbf{p}}; \mathbf{q}) \leq \left| d_{\mathcal{G}}(v^{\mathrm{e}}_{e(\mathbf{p})}, v^{\mathrm{s}}_{e(\mathbf{q})}) + x_{\tilde{\mathbf{p}}} \right| \leq \Delta d_{\mathcal{G}}(\mathbf{p}, \tilde{\mathbf{p}}; \mathbf{q}).$$

Then, the objective function R-VLP, defined by

$$\begin{aligned} & \mathbf{E}\left(\Delta \underline{d}_{\mathcal{G}}\left(P, \tilde{P}; Q\right)\right) \\ =\ & \int\int\int \Delta \underline{d}_{\mathcal{G}}(\mathbf{p}, \tilde{\mathbf{p}}; \mathbf{q}) f_Q(\mathbf{q}) f_{\tilde{P}}(\tilde{\mathbf{p}}|P = \mathbf{p}) f_P(\mathbf{p}) \mathrm{d}\tilde{\mathbf{p}} \mathrm{d}\mathbf{q} \mathrm{d}\mathbf{p} \\ =\ & \sum_i \sum_j c'_{i,j} w_{i,j} \end{aligned} \tag{38}$$

offers a lower bound of $\mathbf{E}\left(\Delta d_{\mathcal{G}}\left(P, \tilde{P}; Q\right)\right)$ as

$$\mathbf{E}\left(\Delta \underline{d}_{\mathcal{G}}\left(P, \tilde{P}; Q\right)\right) - \mathbf{E}\left(\Delta d_{\mathcal{G}}\left(P, \tilde{P}; Q\right)\right) \tag{39}$$

$$\begin{aligned} =\ & \int\int\int \left(\Delta \underline{d}_{\mathcal{G}}(\mathbf{p}, \tilde{\mathbf{p}}; \mathbf{q}) - \Delta d_{\mathcal{G}}(\mathbf{p}, \tilde{\mathbf{p}}; \mathbf{q})\right) \\ \times\ & f_Q(\mathbf{q}) f_{\tilde{P}}(\tilde{\mathbf{p}}|P = \mathbf{p}) f_P(\mathbf{p}) \mathrm{d}\tilde{\mathbf{p}} \mathrm{d}\mathbf{q} \mathrm{d}\mathbf{p} \end{aligned} \tag{40}$$

$$\leq\ 0 \ \ (\text{since } \Delta \underline{d}_{\mathcal{G}}(\mathbf{p}, \tilde{\mathbf{p}}; \mathbf{q}) - \Delta d_{\mathcal{G}}(\mathbf{p}, \tilde{\mathbf{p}}; \mathbf{q}) \leq 0) \tag{41}$$

Note that $\mathbf{E}\left(\Delta \underline{d}_{\mathcal{G}}\left(P, \tilde{P}; Q\right)\right)$ is allowed to be written as $\sum_i \sum_j c'_{i,j} w_{i,j}$ since $\Delta \underline{d}_{\mathcal{G}}(\mathbf{p}, \tilde{\mathbf{p}}; \mathbf{q})$ is relevant to the edges that $\mathbf{p}$ and $\tilde{\mathbf{p}}$ are positioned in, but is irrelevant to $\mathbf{p}$ and $\tilde{\mathbf{p}}$'s specific locations within the edges.

*2) Condition B:* From Equation (13) we can obtain that

$$\int_{e_j} f_{\tilde{P}}(\tilde{\mathbf{p}}|P = \mathbf{p}) \, \mathrm{d}\tilde{\mathbf{p}} \leq e^{\epsilon d_{\mathcal{G}}(e_i, e_l)} \int_{e_j} f_{\tilde{P}}(\tilde{\mathbf{p}}|P = \mathbf{p}) \, \mathrm{d}\tilde{\mathbf{p}}$$

or equivalently,

$$\frac{1}{\int_{e_i} f_P(\mathbf{p}) \, \mathrm{d}\mathbf{p}} w_{i,j} \leq \frac{e^{\epsilon d_{\mathcal{G}}(e_i, e_l)}}{\int_{e_l} f_P(\mathbf{p}) \, \mathrm{d}\mathbf{p}} w_{l,j}. \tag{43}$$

As Equation (43) is a necessary (or relaxed) condition of the Geo-I constraint defined by Equation (13), R-VLP's feasible region $\Omega(\mathbf{W})$, defined by

$$\Omega(\mathbf{W}) = \left\{ \mathbf{W} \left| \begin{array}{l} \frac{1}{\int_{e_i} f_P(\mathbf{p}) \mathrm{d}\mathbf{p}} w_{i,j} \leq \frac{e^{\epsilon d_{\mathcal{G}}(e_i, e_l)}}{\int_{e_l} f_P(\mathbf{p}) \mathrm{d}\mathbf{p}} w_{l,j}, \\ \forall j \text{ and } d_{\mathcal{G}}(e_i, e_l) \leq r. \\ \sum_i \sum_j w_{i,j} = 1 \end{array} \right. \right\}. \tag{44}$$

is a relaxed feasible region for the original VLP. The proof is complete.

■

## REFERENCES


[1] L. Kazemi and C. Shahabi. Geocrowd: Enabling query answering with spatial crowdsourcing. In *Proc. of ACM SIGSPATIAL*, pages 189–198, 2012.

[2] Wei Li, Haiquan Chen, Wei-Shinn Ku, and Xiao Qin. Scalable spatiotemporal crowdsourcing for smart cities based on particle filtering. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '17, pages 63:1–63:4, New York, NY, USA, 2017. ACM.

[3] Yongliang Sun, Yejun Sun, and Kanglian Zhao. *Spatial Crowdsourcing-Based Sensor Node Localization in Internet of Things Environment*, pages 528–536. 01 2018.

[4] Xiao Wang, Xinhu Zheng, Qingpeng Zhang, Tao Wang, and Dayong Shen. Crowdsourcing in its: The state of the work and the networking. *IEEE T-ITS*, 17(6):1596 – 1605, 2016.

[5] Di Wu, Yuan Zhang, Lichun Bao, and Amelia Regan. Location-based crowdsourcing for vehicular communication in hybrid networks. *IEEE T-ITS*, 14:837–846, 2013.

[6] Z. Ou, J. Dong, S. Dong, J. Wu, A. Yla-Jaaski, P. Hui, R. Wang, and A. W. Min. Utilize signal traces from others? a crowdsourcing perspective of energy saving in cellular data communication. *IEEE TMC*, 14(1):194–207, 2015.

[7] A. Misra, A. Gooze, K. Watkins, M. Asad, , and C. A. Le Dantec. Crowdsourcing and its application to transportation data collection and management. *Transportation Research Record: Journal of the Transportation Research Board*, 2414(1):1 – 8, 2014.

[8] H. To, G. Ghinita, L. Fan, and C. Shahabi. Differentially private location protection for worker datasets in spatial crowdsourcing. *IEEE TMC*, pages 934–949, 2017.

[9] Y. Tong, J. She, B. Ding, L. Wang, and L. Chen. Online mobile micro-task allocation in spatial crowdsourcing. In *IEEE ICDE*, pages 49–60, 2016.

[10] Hien To, Cyrus Shahabi, and Li Xiong. Privacy-preserving online task assignment in spatial crowdsourcing with untrusted server. In *Proc. of IEEE ICDE*, 2018.

[11] B. Liu, L. Chen, X. Zhu, Y. Zhang, C. Zhang, and W. Qiu. Protecting location privacy in spatial crowdsourcing using encrypted data. In *EDBT*, 2017.

[12] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proc. of MobiSys*, pages 31–42, 2003.

[13] B. Gedik and L. Liu. Location privacy in mobile systems: A personalized anonymization model. In *IEEE ICDCS*, pages 620–629, 2005.

[14] L. Zheng, H. Yue, Z. Li, X. Pan, M. Wu, and F. Yang. k-anonymity location privacy algorithm based on clustering. *IEEE Access*, 6:28328–28338, 2018.

[15] Chi-Yin Chow. *Cloaking Algorithms for Location Privacy*, pages 93–97. Springer US, Boston, MA, 2008.

[16] Huang Zhangwei and Xin Mingjun. A distributed spatial cloaking protocol for location privacy. *Networks Security, Wireless Communications and Trusted Computing, International Conference on*, 2:468–471, 01 2010.

[17] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE PerCom*, pages 46–55, 2003.

[18] P. Shankar, V. Ganapathy, and L. Iftode. Privately querying location-based services with sybilquery. In *Proc. of UbiComp*, pages 31–40, 2009.

[19] Richard Chow and Philippe Golle. Faking contextual data for fun, profit, and privacy. In *Proc. of ACM WPES*, pages 105–108, 2009.

[20] V. A. Kachore, J. Lakshmi, and S. K. Nandy. Location obfuscation for location data privacy. In *2015 IEEE World Congress on Services*, pages 213–220, June 2015.

[21] J Zhang, K Liu, and J Yang. An obfuscation-based approach for location privacy protection. *Journal of Computational Information Systems*, 9:9781–9789, 12 2013.

[22] M. Li, S. Salinas, A. Thapa, and P. Li. n-cd: A geometric approach to preserving location privacy in location-based services. In *2013 Proceedings IEEE INFOCOM*, pages 3012–3020, April 2013.

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

[24] K. Chatzikokolakis, C. Palamidessi, and M. Stronati. Constructing elastic distinguishability metrics for location privacy. *PoPETs*, 2015:156–170, 2015.

[25] Reza Shokri. Privacy games: Optimal user-centric data obfuscation. *Proceedings on Privacy Enhancing Technologies*, 2015(2):299 – 315, 2015.

[26] L. Yu, L. Liu, and C. Pu. Dynamic differential location privacy with personalized error bounds. In *Proc. of ACM NDSS*, 2017.

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

[28] C. Dwork, , F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*, pages 265–284. Springer Berlin Heidelberg, 2006.

[29] Frederick S. Hillier. *Linear and Nonlinear Programming*. Stanford University, 2008.

[30] Lorenzo Bracciale, Marco Bonola, Pierpaolo Loreti, Giuseppe Bianchi, Raul Amici, and Antonello Rabuffi. CRAWDAD dataset roma/taxi (v. 2014-07-17). Downloaded from https://crawdad.org/roma/taxi/20140717, July 2014.

[31] S. Narain, T. D. Vo-Huu, K. Block, and G. Noubir. Inferring user routes and locations using zero-permission mobile sensors. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 397–413, May 2016.

[32] George Casella and Roger Berger. *Statistical Inference*. Duxbury Resource Center, June 2001.

[33] Harsh Bhasin. *Algorithms: Design and Analysis*. Oxford Univ Press, 2015.

[34] L. Yan, H. Shen, J. Zhao, C. Xu, F. Luo, and C. Qiu. Catcharger: Deploying wireless charging lanes in a metropolitan road network through categorization and clustering of vehicle traffic. In *IEEE INFOCOM*, pages 1–9, 2017.

[35] Cynthia Dwork. Differential privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming*, pages 1–12, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[36] B. Palanisamy, C. Li, and P. Krishnamurthy. Group differential privacy-preserving disclosure of multi-level association graphs. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 2587–2588, June 2017.

[37] F. Ahmed, A. X. Liu, and R. Jin. Social graph publishing with privacy guarantees. In *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, pages 447–456, June 2016.

[38] R. Dewri. Local differential perturbations: Location privacy under approximate knowledge attackers. *IEEE TMC*, 12(12):2360–2372, 2013.

[39] C. Yin, J. Xi, R. Sun, and J. Wang. Location privacy protection based on differential privacy strategy for big data in industrial internet of things. *IEEE Transactions on Industrial Informatics*, 14(8):3628–3636, Aug 2018.

[40] Raluca Ada Popa, Andrew J. Blumberg, Hari Balakrishnan, and Frank H. Li. Privacy and accountability for location-based aggregate statistics. In *Proceedings of the 18th ACM Conference on Computer and Communications Security*, CCS '11, pages 653–666, New York, NY, USA, 2011. ACM.