Optimal privacy control for transport network data sharing

Brian Yueshuai He, Joseph Y. J. Chow*
C2SMART University Transportation Center, New York University, Brooklyn, NY, USA
*email: joseph.chow@nyu.edu

Abstract

In the era of smart cities, Internet of Things, and Mobility-as-a-Service, private operators need to share data with public agencies to support data exchanges for "living lab" ecosystems more than ever before. However, it is still problematic for private operators to share data with the public due to risks to competitive advantages. A privacy control algorithm is proposed to overcome this key obstacle for private operators sharing complex network-oriented data objects. The algorithm is based on information-theoretic k-anonymity and, using tour data as an example, where an operator's data is used in conjunction with performance measure accuracy controls to synthesize a set of alternative tours with diffused probabilities for sampling during a query. The algorithm is proven to converge sublinearly toward a constrained maximum entropy under certain asymptotic conditions with measurable gap. Computational experiments verify the applicability to multi-vehicle fleet tour data; they confirm that reverse engineered parameters from the diffused data result in controllable sampling error; and tests conducted on a set of realistic routing records from travel data in Long Island, NY, demonstrate the use of the methodology from both the adversary and user perspectives.

Keywords: privacy; k-anonymity; open data; tour generation; entropy maximization

To appear in Transportation Research Part C, Special Issue on ISTTT23

1. Introduction

With the development of Internet of Things (IoT), many new transport services are emerging. Just in New York City alone, ride-hail services tripled in ridership in one and a half years and reached 15 million a month (Bloomberg View, 2017). Global car-sharing users increased from 350,000 in 2006 to almost five million in 2014 (Shaheen and Cohen, 2016). These services represent a new transport paradigm: Mobility-as-a-Service (MaaS), and it is becoming increasingly important in modern cities. Numerous studies have argued that for smart cities to thrive, public agencies and private operators need to work together (Djavadian and Chow, 2017; Hensher, 2017; Rasulkhani and Chow, 2019). Indeed, partnerships have sprung up in recent years between public agencies and mobility providers like Lyft, Uber, Via, Car2Go, etc. For example, the Dallas Area Rapid Transit (DART) collaborated with Uber to simplify the connections at transit stations (Jaffe, 2015). These partnerships extend to companies providing services in car sharing, smart parking, incentives programs, real time traffic management, electric vehicle infrastructure provision, personal travel apps, among others. Fundamentally, mobility service provision requires both city agencies and private companies working together. Because of the emergence of MaaS and public-private cooperation, there is a need for data sharing between operators and public agencies or even between multiple operators.

Data sharing can be done in several ways. First, aggregate public data is generally available—for example, the Taxi and Limousine Commission can require for-hire vehicles (FHVs) to share total trip data at certain zonal levels and even companies themselves initiate programs like "Uber Movement" to share average travel times and speeds with the public. Second, operators may share private data with a collaborator in which the results are not shared with the public without some aggregation, typically with a non-disclosure agreement (NDA). The third sharing approach is by an open data exchange. There is an increasing number of online sites serving as "data exchanges" for multiple cities and private operators. One such example is SharedStreets illustrated in Fig. 1; another is the creation of a "Mobility Data Specification" by Los Angeles Department of Transportation (LADOT, 2018; Sadik-Khan, 2019).

This last approach is where we envision our proposed method to address. Open data exchanges are critical for supporting data-driven innovations in "living lab" ecosystems (Schaffers et al., 2011). For example, for entrepreneurs interested in creating parking apps, they would need realistic parking occupancy and inventory data to test their algorithms against. Open data exchanges are also important for public agencies to provide decision support for their public services, which are becoming increasingly multi-stakeholder, interoperable (Colpaert et al., 2014), and information-centric (Piro et al., 2014). For example, a city-operated Mobility-as-a-Service operation can involve multiple partners: multiple transit operators, a smart grid provider, a fare manager, a mobile app provider, among others. One might argue that the second approach of signing NDAs would suffice for sharing data in this arrangement. However, many of these operations require the public agencies to facilitate multi-stakeholder operations. Even with the NDA requiring providers not to share their data with other third parties, it is hard to control information sharing between natural competitors to ensure interoperability. In general, there is a problem of designing mechanisms to make it easier for private mobility providers to share their data with public agencies and with each other without significantly compromising competitiveness.

This concept of data-driven innovations of "living lab" ecosystems will only succeed if operators are willing to share their operational data with public agencies and researchers. Convincing private operators to share data remains a major obstacle (Janssen et al., 2012). For

example, private mobility companies resist sharing their route data, instead offering in limited cases some passenger pickup information (e.g. FiveThirtyEight, 2016). In other cases, information is shared in such an aggregated form that it is rendered useless for high resolution analysis. An example is truck GPS data, which operators are generally unwilling to share publicly. Sharing a sample of such data from multiple carriers operating in a city like New York would help support urban freight policies.

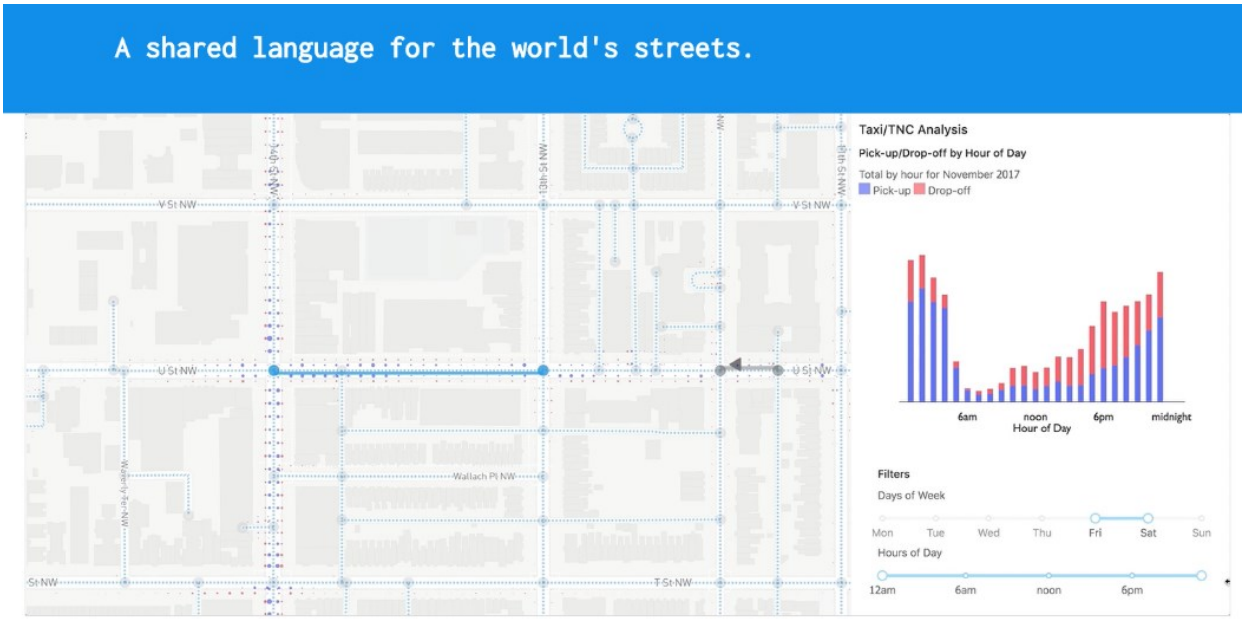
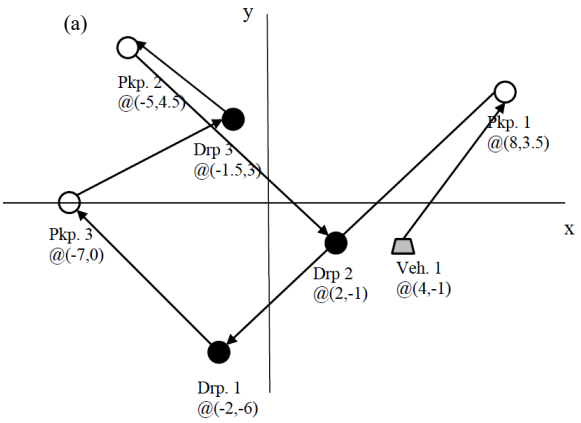


Fig. 1. SharedStreets, a non-profit digital commons and clearinghouse for data exchange toward public-private collaboration (www.sharedstreets.io).

The reluctance to share more complete information with the public, e.g. complete breadcrumb data or transaction timestamps, makes sense from the private operator's perspective. Such data, if exposed to adversaries, can be used to reverse engineer the operator's algorithms to steal competitive algorithm designs and policies. Note that this is a different type of attack than a cybersecurity threat (as illustrated by Yuan et al., 2016, for MaaS). We illustrate this reverse engineering quantitatively with an example using shared taxi service, although other examples like truck GPS data or location-based service microtransactions also apply.

Consider a route observed in Fig. 2(a) for a 2-passenger capacity shared taxi, where a vehicle is observed to pick up and then drop off passengers no. 1, 3, and then 2 in that order. Given the passenger capacity, the vehicle could have instead chosen to take the following path: $(0,3P, 2P, 3D, 2D, 1P, 1D)$, where 0 is the vehicle's location, and a P designation is for pickup and a D is for drop-off corresponding to the passenger. The routing algorithm and policies governing that algorithm are guarded secrets. We can descriptively fit a general mixed integer programming (MIP) structure of a Dial-a-Ride Problem (DARP) to that policy, as shown in Fig. 2(b), where the constraint specifications and parameters (including objective weights γ, α, β pertaining to travel cost, ride time, and wait time) need to be estimated to fit any open tour (route) policy in practice. These are relative measures; i.e. $\gamma = 2, \alpha = 1, \beta = 1$ yield the same results as $\gamma = 4, \alpha = 2, \beta = 2$. To allow for operators that might impose a weight of zero to one or more of the objectives, we have to specify all three weights.

The model in Fig. 2(b) is a standard formulation from the literature and can be found in a less parametric form in Cordeau and Laporte (2007) so we only explain the terminology. The $P = \{1, \dots, n\}$ is the set of pickup locations, $D = \{n+1, \dots, 2n\}$ is the set of drop-off locations, V is fleet size, vehicle "depots" are $\{0_1, \dots 0_{|V|}, 2n+1\}$, N is the set of all nodes, $q_i, i \in P$, is the group size, d_i is the service duration (loading/unloading/waiting), u is vehicle capacity, c_{ij} is the travel cost, t_{ij} is the travel time, R_{max} is the maximum ride time, X_{ijk} is the route decision of vehicle k, $T_i, i \in \{P, D\}$, is the start of service at node i, W_i is the load upon leaving node i, and $R_i, i \in P$, is the ride time of pickup i. Note that the Cordeau and Laporte (2007) formulation includes time windows as well, but as that is unobservable from cross-sectional route data we leave those constraints out as unidentifiable. If panel data is available then it is possible to infer time windows as well. For this study we assume only cross-sectional data is available without loss of generality. An adversary would then seek to learn the parameters used by the operator based on the observations of the routes.



If uninformed objective weight priors, i.e. $\alpha_0 = \beta_0 = \gamma_0 = 1$, route would have been $(0, 3P, 2P, 3D, 2D, 1P, 1D)$. Instead, we observe $(0, 1P, 1D, 3P, 3D, 2P, 2D)$ (arrows shown).

(b)

$$min\ Z = \gamma \sum_{k\in V}\sum_{i\in N}\sum_{j\in N} c_{ij}X_{ijk} + \alpha \sum_{i\in P} R_i + \beta \sum_{i\in P} T_i$$

Subject to

$$\sum_{k\in V}\sum_{j\in N} X_{ijk} = 1,\ \forall i \in P$$

$$\sum_{j\in N} X_{0_k jk} = \sum_{j\in N} X_{j,2n+1,k},\ \forall k \in V$$

$$\sum_{j\in N} X_{0_k jk} \le 1,\ \forall k \in V$$

$$\sum_{j\in N} X_{ijk} = \sum_{j\in N} X_{n+i,jk},\ \forall i \in P, k \in V$$

$$\sum_{j\in N} X_{jik} = \sum_{j\in N} X_{ijk},\ \forall i \in P \cup D, k \in V$$

$$T_i - T_j \le -d_i - t_{ij} + (1 - X_{ijk})M,\ \forall i, j \in N, k \in V$$

$$W_i - W_j \le -q_j + (1 - X_{ijk})M,\ \forall i, j \in N, k \in V$$

$$T_{n+i} - T_i - R_i \le d_i,\ \forall i \in P$$

$$0 \le W_i \le u,\ \forall i \in N$$

$$X_{ijk} \in \{0,1\}$$

$$t_{i,n+1} \le R_i \le R_{max}$$

$$0 \le T_i \le T_{max}$$

Fig. 2. (a) An observed open tour, and (b) a MIP formulation for a generic open tour DARP.

Machine learning techniques designed for network optimization models can be used to learn the parameters of the routing policy. One such technique, for example, is inverse optimization (Ahuja and Orlin, 2001; Wang, 2009; Xu et al., 2018), which we discuss further in Section 2. We can readily show that solving an inverse optimization of the MIP based on L_1-norm minimization from uninformed priors of $\alpha_0 = \beta_0 = \gamma_0 = 1$ using the cutting plane method from Wang (2009) would converge in four iterations to an optimal solution in which $\alpha^* = 2.1255$, $\beta^* = 0.0245$, and $\gamma^* = 1$ with a MIP objective value of 110.195. By using a single sample of a vehicle's trajectory data of pickups and drop-offs, a competitor can guess that this operator values passenger ride time highly compared to wait time, and has an effective objective value of 110.195 compared to a value of 151.642 under uninformed priors. From there, a competitor can further test different constraints

or policies to see how they fare relative to the gap between the prior objective value and the effective objective value from the observed data. Additional route data would further improve the efficiency of this adversarial attack.

Tour data can potentially be used by an adversary to infer the following information about the operator or their users:

- Identification and prioritization of different routing objectives, e.g. travel times or route length, passenger wait times, passenger total journey times, or vehicle utilization
- Identification of dispatch criteria and constraints
- Existence and value of hard time windows or penalties for soft time windows (under panel data setting)
- Importance placed on minimizing future costs in a dynamic algorithm
- Presence and value of constraints to limit amount of passenger detours
- Value of destinations in profitable tour problems in which destinations are chosen among a set of candidates

Clearly, operators would not willingly share their data with an open data exchange unless their privacy was protected. Researchers have looked at this type of problem for over a decade. As suggested by Abowd and Lane (2004) and Dwork (2006), data privacy may be achieved in several ways, most of which involve the generation of synthetic noise. The crux of that research has either focused on user privacy, which differs from operator privacy, or it has not considered the complexity of network-oriented data objects like synthetic tours. The problem of constructing a synthetic route that (1) is representative of a real route, (2) provides sufficiently useful information to a public agency user, and (3) is noisy enough to confuse an adversarial attack has not been studied.

Formally, our research problem is stated as follows. We seek a privacy control mechanism that can take *network-oriented* data objects in a data exchange and respond to data queries with synthetic data objects that:

1) Are feasible network solutions;
2) Would be, on average over multiple queries, sufficiently similar to the real data object with regard to a performance measure specified in the query;
3) The diffusion of the real data object to the synthetic data objects maximizes the anonymity of the real object among the synthetic data objects.

The first condition means that a synthetic object should look real; there should not be telltale signs in which an adversary can automatically remove the object, for example if a passenger is shown to be dropped off before being picked up. For the second condition, a query should be related to a performance measure of interest. For example, trip data might be queried because the data user making the query is either interested in (a) origin location distributions or (b) OD travel time distributions. Depending on the measure, the control should respond with the appropriate synthetic data. For the third condition, an observer should not be able to discern the real data object from synthetic data objects based on the diffusion probabilities.

The proposed privacy control mechanism design is illustrated in Fig. 3. Operators provide historical operation data as the input. When a user (an open data researcher or a collaborator in the multi-operator MaaS system) queries data, they receive randomly synthetized data in order to protect the privacy of the operator. The synthetic data needs to be carefully designed to capture

aspects specified by the user sufficiently accurately while ensuring that the retrieved data cannot be easily used to reverse engineer the operator's policies.

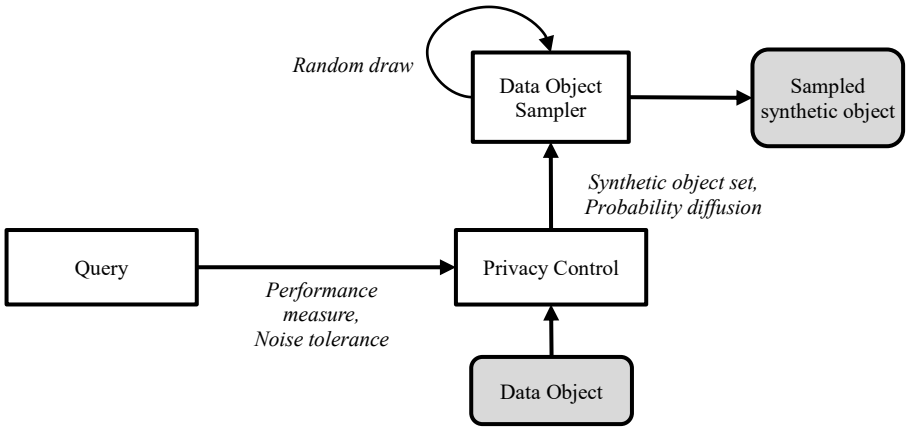


Fig. 3. Privacy control mechanism design for data exchanges.

In this study we test the mechanism on the criterion of having similar passenger journey times, but other criteria can also be used: location, time of pickup, vehicle identification, etc. In essence, this method introduces more control in how network-structured private data can be shared.

We propose a privacy control mechanism based on "κ-anonymous diffusion". For a given data object and a query's desired performance measure and tolerance, the control produces a set of up to κ synthetic data objects with assigned probabilities of querying each object. This method can be applied to numerous types of network-oriented data objects, including vehicle assignment decisions, scheduling decisions involving continuous time variables, route/tour decisions, location decisions, etc. Throughout this study we focus on the open tour version of the DARP without time windows as shown in Fig. 2(b) because it falls on the complex end of the spectrum and any insights should be valid for simpler data objects. Furthermore, the methodology should be applicable to data exchanges beyond MaaS, where data objects may involve network elements: e.g. urban freight (private truck GPS data), smart grid (competing energy providers setting locational marginal prices), counterterrorism (geospatial intelligence data), cybersecurity (network security protocols), and social networks (social contacts' data).

The rest of the paper is organized as follows. Section 2 reviews the related research about privacy control. Section 3 proposes the methodology that we adopt to find an optimal privacy control mechanism. Numerical verification and a case study based on a large-scale simulated scenario drawn from real data in Long Island, NY, are reported in Section 4. Section 5 concludes the study.

2. Literature review

2.1. Prior studies

Privacy control has become a well-established field in the last two decades. Privacy control methods deal with aggregating data or synthesizing data to create additional noise. However, much of the attention has focused on user privacy. For users, sharing data with the public makes it easier for their identities to be stolen or for personal information to be made available to the wrong

people. One example of user privacy concerns is demonstrated with taxi data. Trotter (2014) shows how a combination of paparazzi photos and taxi breadcrumb data can reveal unsettling amounts of personal detail: origin/destination of the trip, fare paid, and tip amount.

One of the prevailing methods in user privacy control is differential privacy (Dwork, 2006, 2008). Differential privacy involves applying a function $\mathcal{K}$ to a database that randomizes the data in such a way that the aggregate data output has at most ϵ difference when one element of the database is removed. By ensuring this, users can participate in the database without fear of being identified because the difference between the filtered databases with and without their data would be nominal (less than ϵ). There is a wide range of applications of differential privacy in transport: Chen et al. (2012), Kargl et al. (2013), Le Ny and Pappas (2014), and Dong et al. (2015), among others.

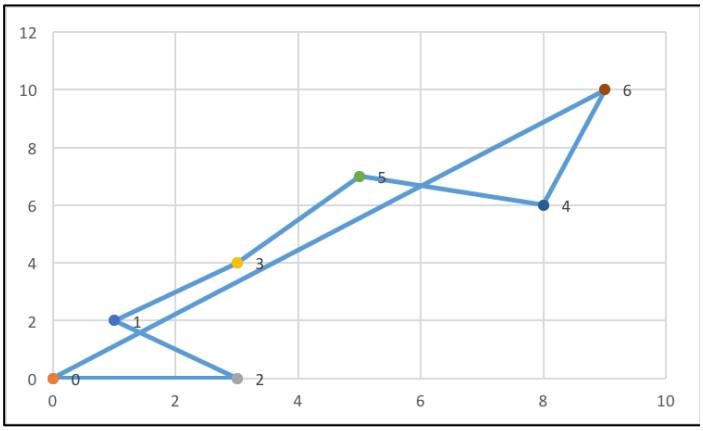
In the case of operators, the concern is the risk of even their operation strategies being reverse engineered by adversaries from the data they share. This risk exists even for a single observation of a data object because each can uniquely inform on the underlying policies. Unlike user privacy, the objective is not to "hide in a herd" because there is no herd to hide in. The objective is therefore to limit the amount or certainty of information shared to minimize this risk. This trade-off leads to an information control problem (Sankar et al., 2011; Dong et al., 2015; Belletti and Bayen, 2017). In the case of Belletti and Bayen (2017), for example, they formulated a model to limit information sharing without compromising operability in the case of matching MaaS fleet drivers with passengers.

In general cases, limiting data can be done by providing it with noise created around it. An example of this type of privacy control is in Tsai et al. (2015) and Wang et al. (2017). The authors introduced the concept of κ-shortest path privacy in which a network's link costs are perturbed minimally such that at least κ shortest paths between given origin and destination vertices are identical in length. The κ-anonymous, information-theoretic framework (Sweeney, 2002) provides uniform diffusion of a data object in the sense that each of the K objects is equally likely to occur to an outsider observing this perturbed network. This approach allows a whole network to be shared while protecting the identity of its shortest path. However, in the case of transport networks, the link weights are often observable so this approach of κ-anonymity is not applicable. He et al. (2017) addressed this research gap by proposing an alternative way of sharing network data objects.

2.2. Overview of He et al. (2017)

In a conference paper, the authors showed that the optimal diffusion of a data object into a set of κ synthetic data objects (such as a set of tours) for querying randomly can be modeled as an entropy maximizing convex optimization program (see Sun et al., 2013). If there are no constraints in diffusing the data object, the query probabilities would converge toward $1/\kappa$ for each synthetic object. This makes sense because a discrete uniform diffusion exhibits the highest anonymity in the set. As an example, consider diffusing a single-vehicle tour $(0,2,1,3,5,4,6,0)$, where $\{1,2,3\}$ are pickup locations of three passengers and $\{4,5,6\}$ are corresponding drop-offs. The tour is shown in Fig. 4(a) with details of the arrival times in Fig. 4(b). If the desired performance measure is average passenger travel time and there are only 90 feasible tours based on explicit enumeration, then the optimal diffusion with an average passenger arrival time error tolerance of $\Delta = 0.1$ is shown in Fig. 5. The solution in Fig. 5 shows that the $\Delta = 0.1$ tolerance is a binding constraint since different tours have different query probabilities while one cannot discern the true tour (first tour in Fig. 5) from those probabilities.

(a)



(b)

Node	Arrival Time
0	0
2	3
1	5.828
3	8.657
5	12.262
4	15.425
6	19.548
0	33.001
Passenger	**True travel time**
Passenger 1 true travel time $(t_4 - t_1)$	9.597
Passenger 2 true travel time $(t_5 - t_2)$	9.262
Passenger 3 true travel time $(t_6 - t_3)$	10.892

Fig. 4. (a) A tour data object, and (b) arrival time details of that tour (He et al., 2017).

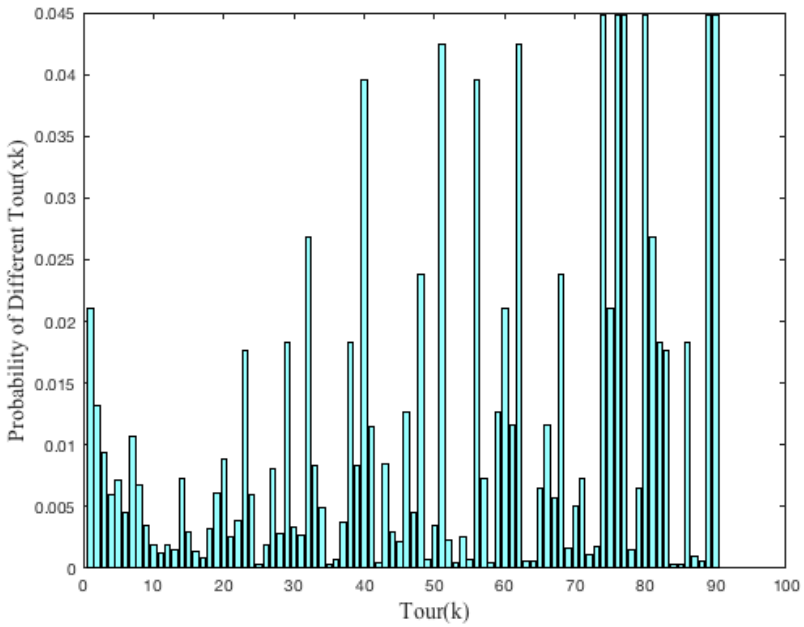


Fig. 5. Optimal probabilities assigned to tour set (He et al., 2017).

Several research gaps associated with the methodology can be identified from this example. In practice, it is not possible to enumerate all possible tours because of its combinatorial nature. There needs to be a way of efficiently generating data objects. For example, if only $\kappa = 10$ tours were requested for this example, which ten tours should be generated? Some of the tours in Fig. 5 are shown to exhibit probabilities of zero. Having them in the set would be useless because they would never be used. Similarly, tours that result in all zero (or infinite) valued objective coefficients from inverse optimization suggests they are not realistic and should also not be included. Tours need to be selected such that maximum anonymity is achieved as defined by

maximum entropy diffusion of probabilities of drawing each tour from the set in a query. For example, if the subset of first 10 tours in Fig. 5 were used as opposed to a subset of the last 10 tours in the set, the achievable entropy can be different. This is verified with experiments in He et al. (2017). A tour generation algorithm needs to optimally select a set of tours to maximize anonymity.

In the current study, we tackle these significant issues and propose a heuristic algorithm for privacy control for network-oriented data sharing. The algorithm is shown to be exact when the set of data objects is equal to the full enumerated set, and a gap can be quantified from smaller subsets using a relaxed constraint upper bound gap. Before presenting the algorithm, we provide a brief overview of inverse optimization as it is used to represent an adversarial attack.

2.3. Overview of inverse optimization

Inverse optimization (IO) is a parameter estimation methodology to align the optimal solutions of optimization models with observed outputs. A classic example is the inverse shortest path problem (Burton and Toint, 1992), where the link costs of a network are minimally perturbed from priors so that an observed path would be optimal. Ahuja and Orlin (2001) generalized the inverse optimization for linear programming (LP) problems. Consider an LP under matrix notation: $\min\{c^T x: Ax \leq b, x \geq 0\}$. The n-dimensional vector x is a set of decision variables, c is an n-dimensional vector of objective coefficients, A is an $m \times n$ constraint matrix, and b is an m-dimensional vector of side constraint values. The inverse optimal set of parameters for a given set of objective coefficient priors c_0 and observed decision vector x^* under L_1-norm is obtained with the problem in Eq. (1). The L_1-norm minimization from a prior can provide regularity (Tenorio, 2001) for the inverse problems. This ensures that a solution can be obtained given a prior c_0. For example, even though objective weights for the DARP in Fig. 2(b) are relative ($\gamma = 2, \alpha = 1, \beta = 1$ is equivalent to $\gamma = 4, \alpha = 2, \beta = 2$), the IO problem is formulated to perturb minimally from a prior to obtain a unique solution. Ahuja and Orlin (2001) showed that the inverse LP can be reformulated as an LP.

$$\min_{c} \ |c_0 - c|: x^* = argmin\{c^T x: Ax \leq b, x \geq 0\} \tag{1}$$

There have been several advances and applications in IO. Xu et al. (2018) provide a summary of these advances. Among the applications, Xu et al. (2018) showed how network properties can be inferred by an external observer. Birge et al. (2017) used IO to monitor external influences on an electricity market. Chan et al. (2014) inferred revealed objective weights of a multiobjective program with unknown objective preferences to help cancer therapy. These applications suggest that IO can be an effective inference methodology, even for malevolent purposes.

In the case of inverse integer programs (IPs), Wang (2009) proposed a cutting plane algorithm to solve the inverse IP as a series of LPs, which has been applied to vehicle routing problems (Chow and Recker, 2012; You et al., 2016). This is the method we use on the synthetic tours to determine the parameters corresponding to them. For a sample of synthetic tours, the inverse IP parameters should have maximum standard errors.

3. Proposed algorithm

3.1. κ-anonymous information-theoretic algorithmic framework for tours

As an information-theoretic privacy control mechanism, the objective of anonymity is set equivalent to entropy maximization of query probabilities assigned to synthetic objects. A generalized formulation is presented in Eq. (2) for tours as the data objects, which can be readily adapted to other data objects.

$$\max_{K,x_k} E = -\sum_{k\in K} x_k \ln x_k \tag{2a}$$

Subject to

$$\sum_{k\in K} x_k \le |V| \tag{2b}$$

$$\sum_{k\in K} \delta_{jk} x_k \ge 1, \forall j \in O \tag{2c}$$

$$S(\{x_k, r_k\}, \Delta) \le 0 \tag{2d}$$

$$0 \le x_k \le 1 \tag{2e}$$

$$|K| = \kappa, \qquad K \subset \Omega \tag{2f}$$

where Δ is a desired accuracy tolerance for a specified metric (e.g. travelers' ride times), S is a constraint set corresponding to that metric, r_k is a tour from which metrics can be derived as parameters (e.g. travelers' ride times), and δ_{jk} is set to 1 when node j is visited by tour k, and 0 otherwise. For example, a tour may be a sequence of 2-tuples: $r_k = [(0,0),(2,10),(4,25),(1,30),(3,45)]$, where each 2-tuple is a $(u,\ t_{uk})$ with arrival time t_{uk} at node u, $\{0\}$ is the vehicle's initial location, $\{1,2\}$ are the pickup locations of two passengers, and $\{3,4\}$ are their corresponding drop-off locations. This tour informs us that passenger 2 had a ride time of $t_{4k} - t_{2k} = 25 - 10 = 15$ minutes. The value Δ would be negotiated between the operator and the public agency; the operator will generally want this value to be higher while the agency will want a lower value. Queries from users will abide by this agreed upon value. Entropy maximization of the probabilities x_k of a set of tours $r_k, k \in K$, is achieved with objective (2a) and constraints (2b) – (2f), where $K \subset \Omega$ is an endogenous subset of all feasible tours Ω. Solving Eq. (2) requires determining $K \subset \Omega$ that maximizes Eq. (2a).

Here the definition of the problem differs from He et al. (2017) regarding the endogeneity of K and the constraints (2d). He et al. (2017) specify the constraints of the performance metric to be travelers' ride times in constraint set (2d), whereas in this study we use a more general formulation applicable to other data objects.

When unconstrained, the objective value reaches the maximum when all objects have the same likelihood. For example, if $\kappa = 3$, the maximum entropy $E^* = 1.099$ with $x_1^* = x_2^* = x_3^* = 1/3$. This unconstrained solution serves as a constraint-relaxed upper bound for Eq. (2).

There are two primary components to the κ-anonymous privacy control mechanism: (1) generating a set of κ data objects and (2) constructing a query probability filter to maximize the

anonymity of the real object among all synthetic ones. We can trivially see that a master problem with these two components can be decomposed into two subproblems; i.e. if we can find an entropy maximizing set of up to κ data objects and, sequentially, obtain an entropy maximizing set of probabilities associated with each of the given data objects, this solution would be optimal.

Let us call the two components SP1 and SP2. SP1 generates a new tour $\{r_k\}_{k\leq\kappa}$. A loop can be used to call SP1 repeatedly until a set of κ tours are generated. SP2 is then called to assign a probability diffusion $\{x_k\}_{k\leq\kappa}$ of the real object to each object in the generated κ objects $\{r_k\}_{k\leq\kappa}$. Each iteration, SP1 generates a new tour r_k to add to a dynamic set $\widetilde{K}$. The structure is ideally a nested one; SP1 would obtain all the tours that would be assigned a non-zero probability in the entropy maximization step in SP2. An example of a structure without iteration would be an SP1 that explicitly enumerates every tour, running SP2 to assign probabilities, and removing the subset of tours that do not have any probabilities assigned. However, SP1 is based on implicit enumeration and not guaranteed to find such a set that is fully assigned probabilities; some of the tours found may end up unused in SP2. As a result, those unused routes are removed from the list and SP1 is run again to find additional tours to fill in the remainder. The proposed iterative solution framework is shown in Fig. 6.

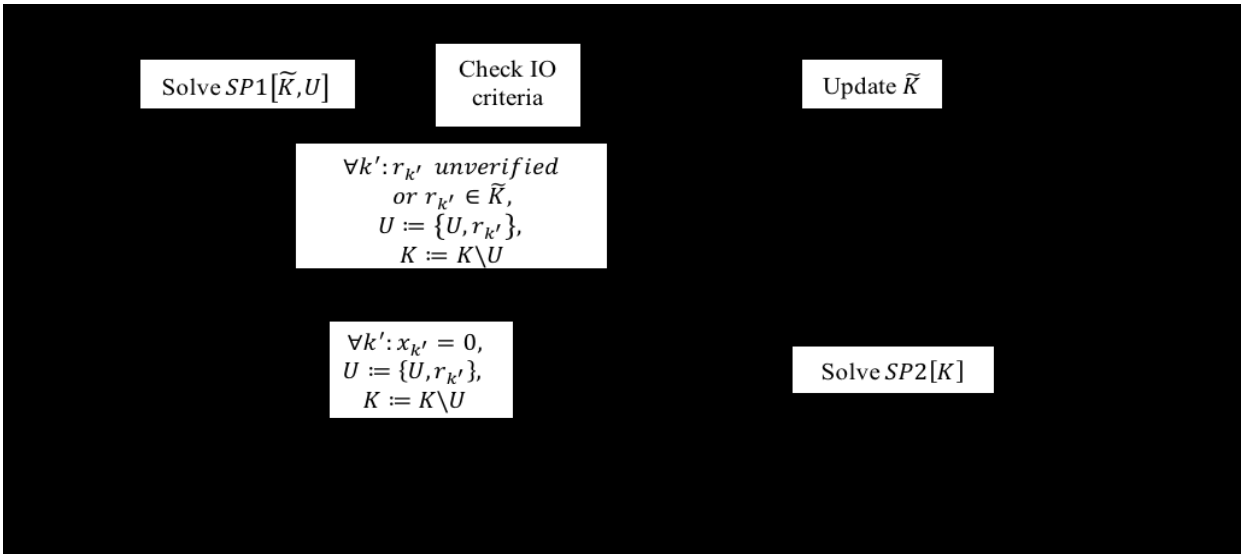


Fig. 6. Algorithm for κ-anonymous privacy control mechanism.

Aside from the two functions for solving SP1 and SP2, other functions are used to ensure that infeasible tours—those that result in unrealistic IO inferences (via IO check) or in tours with zero probability diffusion—are removed and stored in a "undesired" set U. This undesired set is used along with the dynamic set $\widetilde{K}$ to ensure that deleted tours are discouraged while new tours are sought.

As discussed in Section 1, the ride-sharing tour decision can be setup as a DARP. The IO can be used to infer the weights α, β, γ from observed tours. Denote $x^* = [X^*, T^*, W^*, R^*]$ to represent the true tour data. The object of IO for the tour data is shown in Eq. (3).

$$\min_{\alpha,\beta,\gamma}|\alpha-\alpha_0|+|\beta-\beta_0|+|\gamma-\gamma_0|: x^* = argmin\left\{\gamma * \sum_{k\in V}\sum_{i\in N}\sum_{j\in N} c_{ij}x_{ijk} + \alpha * \sum_{i\in P} R_i + \beta * \sum_{i\in P} T_i : Ax \le b, x \ge 0, x \in \mathbb{Z}^+\right\} \quad (3)$$

where A is the constraint matrix for the DARP and b is the original right side vector. By defining $\alpha_0 - \alpha = e_\alpha - f_\alpha$, $\beta_0 - \beta = e_\beta - f_\beta$, and $\gamma_0 - \gamma = e_\gamma - f_\gamma$, where the $\{e, f\}$ vectors are non-negative variables, we can initiate Algorithm 1 to estimate posterior coefficients. The IO inference is used strictly as a constraint check; there is no guarantee in this mechanism that the entropy maximization is equivalent to maximizing IO error (and we do not make such an assertion).

Algorithm 1. Wang's (2009) cutting plan algorithm for inverse IP

Inputs: observed decision variables of original IP x^*, parameters (A, b, I) of IP $\max_{x}\{c^T x: Ax \le b, x \ge 0, x_i \in \mathbb{Z}, \forall i \in I\}$, prior objective coefficients c_0

0. Initiate an empty set $\mathcal{S} = \{\}$ of constraints.
1. Solve Eq. (4) to Eq. (7) and let (y^*, e^*, f^*) be an optimal solution (for minimization IPs Eq. (6) would be "less than or equal to").

$$\min_{y,e,f} w^T e + w^T f \quad (4)$$

Subject to

$$A^T y \ge c_0 - e + f \quad (5)$$

$$(c_0 - e + f)^T x^* \ge (c_0 - e + f)^T \tilde{x}^s, \qquad \forall \tilde{x}^s \in \mathcal{S} \quad (6)$$

$$y, e, f \ge 0 \quad (7)$$

Note: other constraints may be added to this to ensure that the constraints in the original IP are met. For example, if $c \ge 0$ is needed to work, then the inverse problem should include constraints $c_0 - e + f \ge 0$.

2. $\tilde{x} = \operatorname{argmax}_{x}\{(c_0 - e^* + f^*)^T x: Ax \le b, x \ge 0, x_i \in \mathbb{Z}, \forall i \in I\}$. If $(c_0 - e^* + f^*)^T x^* \ge (c_0 - e^* + f^*)^T \tilde{x}$, then stop, and $c^* = c_0 - e^* + f^*$. Otherwise, $\mathcal{S} \coloneqq \mathcal{S} \cup \{\tilde{x}\}$ and go to Step 1.

Outputs: Estimated posterior objective coefficients c^*

When SP1 selects a synthetic tour to generate, the "Check IO criteria" solves an inverse IP with Algorithm 1. If $\alpha_k + \beta_k + \gamma_k = 0$, this means an IO inference would suggest the tour is not realistic and it is then discarded into the U pool. The IO criteria may be "switched off" as well if the IO optimal parameters are not an issue to consider. Note also that the first tour when $k = 1$ is the true tour.

This algorithm converges to a constrained maximum entropy solution if SP1 and SP2 combine to find an entropy maximizing diffusion and tour set. The subproblems are explored in greater detail to show that this algorithm should be convergent under certain conditions.

3.2. Subproblem SP2: probability diffusion with "passenger ride time" accuracy

SP2 takes a set K of tours and assigns probabilities to each tour to maximize the entropy objective in Eq. (2a) while ensuring that a performance measure tolerance is met. In this case, we define a performance measure of "passenger ride time" $(t_{j+n,k} - t_{jk})$ in the synthetic tour r_k and compare that to the observed time τ_j to ensure that the expected relative difference does not exceed Δ. Objective (2a) with exogenous K is subject to constraints in Eq. (8) to Eq. (12), where Eq. (10) – (11) represent the passenger ride time specification for Eq. (2d). Since this subproblem is given a set K, Eq. (2f) is not needed.

$$\sum_{k \in K} x_k \leq |V| \tag{8}$$

$$\sum_{k \in K} \delta_{jk} x_k \geq 1, \forall j \in O \tag{9}$$

$$\sum_{k \in K} \left(t_{j+n,k} - t_{jk}\right) x_k - \tau_j \sum_{k \epsilon K} x_k \leq \Delta \tau_j \sum_{k \epsilon K} x_k \,, \forall j \in O \tag{10}$$

$$-\sum_{k \in K} \left(t_{j+n,k} - t_{jk}\right) x_k + \tau_j \sum_{k \epsilon K} x_k \leq \Delta \tau_j \sum_{k \epsilon K} x_k \,, \forall j \in O \tag{11}$$

$$0 \leq x_k \leq 1 \tag{12}$$

where δ_{jk} is 1 when node j is visited by tour k, otherwise 0; and t_{jk} is the arrival time at node j via tour k. The problem defined by Eq. (2a) and (8) to (12) is a concave optimization program with linear constraints. The only decision variables are the probabilities x_k. The entropy maximization is known to be concave (see Wilson, 1967; He et al., 2017). As a result, any convex optimization algorithm can obtain a global optimum. For convenience we employ a Frank-Wolfe algorithm (Frank and Wolfe, 1956) to obtain the maximum entropy diffusion.

3.3. Subproblem SP1: tour generation

We propose a tour generation subproblem that solves a DARP as shown in Fig. 2(b), but modified such that the link costs in the objective function are constructed from the count of tours that have been visited beforehand, i.e. $\widetilde{K} = \{K, U\}$. To illustrate this point, consider some n^{th} iteration for a 4-node tour generation problem. Suppose the last iteration results in 3 tours forming K and 2 tours discarded in U:

$$K = \begin{Bmatrix} (0,1,2,3,4,0) \\ (0,2,4,3,1,0) \\ (0,2,1,3,4,0) \end{Bmatrix}, U = \begin{Bmatrix} (0,3,2,4,1,0) \\ (0,3,1,4,2,0) \end{Bmatrix}$$

In this case, the new cost matrix to be used in SP1 in the n[th] iteration should be as shown in Table 1. The route sequence obtained from the modified DARP is set as the candidate tour from SP1. Other tour or routing problems should be able to work with this methodology as well.

Table 1. Illustration of constructed cost matrix for SP1

c_{ij}^n	0	1	2	3	4
0	0	1	2	2	0
1	2	0	1	1	1
2	1	1	0	1	2
3	0	2	1	0	2
4	2	1	1	1	0

The algorithm in Fig. 6 based on SP1 and SP2 is not guaranteed to converge to the constrained maximum entropy problem in general. As a special case, when κ is sufficiently large (essentially covering all feasible data objects) or $\Delta \geq D$ (where D is a sufficiently large number), it can be shown that the proposed algorithm would converge to the exact solution. A sketch of the proof is as follows. There are two cases of the problem to consider: tours without overlapping links (independent tours), and tours with overlapping links; for each case, we consider sufficiently large κ or Δ for a total of four cases. For independent tours, let us consider $\kappa = |\Omega|$ with constraint set S: $\sum_k x_k g_{kj} \leq \Delta$ where g_{kj} is a generic performance measure for passenger j in tour r_k and x_k is the query probability for tour r_k. Eq. (10) – (11) fall under this structure with $g_{k,j} = \frac{|(t_{j+n,k} - t_{jk}) - \tau_k|}{\tau_k}$. For the proposed algorithm, initially all tour costs are set to $c_k = 0$. Since there is no overlap, each inner loop of SP1 would generate a new tour that has not yet been chosen until all tour costs are $c_k = 1$ in $\kappa = |\Omega|$ iterations. For the case where $\kappa < |\Omega|$, when $\Delta \geq D$ the difference in entropy between $\{1, 2, 3\}$ and $\{3, 4, 5\}$ reduces toward 0 (the unconstrained case). This means convergence to Eq. (2) is attainable by either of two conditions: when $\kappa \to |\Omega|$ or when $\Delta \geq D$.

In the case where multiple tours share links, selecting one tour can mean adding to the cost for other tours that share links. In the case of $\kappa = |\Omega|$, the algorithm will find all tours since differences in cost functions between a tour $k \in U$ and an overlapping tour $k' \notin U$, $|c_k - c_{k'}|$, increases monotonically until tour k' is an optimal solution to SP1. For $\kappa < |\Omega|$ and $\Delta \geq D$, the $\Delta \geq D$ reduces the difference in entropy in the same way rendering the differences to be like the unconstrained case. In this way, the same conditions apply, although the rate of convergence may be much slower than in the case of completely independent tours.

3.4. Algorithm performance

Three insights of this algorithm need to be discussed. First, the rate of convergence of this algorithm, unfortunately, is not very efficient as indicated by Proposition 1. Its efficiency is on par with the well-known Method of Successive Averages.

Proposition 1. *The algorithm in Fig. 6 has a sublinear rate of convergence.*

Proof.
This can be verified by the cost function. By design, for a fixed set of n tours the algorithm converges when every tour is found. In the fastest scenario, each tour would be identified once so each tour would contribute a weight to the cost function of $\frac{1}{n}$. Assuming the total number of tours approaches infinity, then the weight trivially converges toward 0 in order to find all the tours. Sublinear convergence rate for a sequence $\{x_n\}$ toward L occurs if $\lim_{n\to\infty} \frac{|x_{n+1}-L|}{|x_n-L|} = 1$. In this case, $x_n = \frac{1}{n}$ and $L = 0$. Therefore, $\lim_{n\to\infty} \frac{\left|\frac{1}{n+1}\right|}{\left|\frac{1}{n}\right|} = \lim_{n\to\infty} \frac{n}{n+1} = 1$. ■

It can take many iterations to find a next feasible tour to add to the set, which is an issue that we experienced in some cases. Nonetheless, this brings us to the second insight. Since the algorithm is seeking a constrained maximum entropy, the maximum entropy with relaxed accuracy tolerance constraints provides an upper bound to the optimum. For a given κ it is easy to compute as $\hat{E}_\kappa$ shown in Eq. (13). This means any solution at any iteration can be evaluated for a gap relative to this upper bound, which we call an "upper bound gap".

$$\hat{E}_\kappa = -\ln\left(\frac{1}{\kappa}\right) \tag{13}$$

Third, the upper bound gap can be used as a stopping condition. In the future, modifications will be investigated to speed up this algorithm.

The performance of the algorithm can also be assessed from the perspective of the adversary. Once an operator defines a set of probabilities x_k for a set of tours K based on entropy maximization, an adversary can query the system repeatedly to sample the synthetic tours and solve the IO for each tour to obtain estimates $\hat{\theta}$ for the parameters $\theta = \{\alpha, \beta, \gamma\}$. Based on the sample, the adversary's standard error for each $\hat{\theta}$ can serve as a measure of the algorithm performance; a higher adversary IO standard error (which differs from the operator's entropy maximization probability distributions) reflects less reliability in using IO to infer the parameters. To further understand the proposed algorithm, a series of computational experiments are next conducted.

4. Computational experiments

Several experiments are conducted, first on two toy examples, one involving a single vehicle fleet and another with a two-vehicle fleet, and then a computational case study using simulated tours derived from real data in Long Island, New York. The first set of experiments on the toy instances are performed to verify and illustrate the proposed privacy control mechanism for single- and multi-vehicle fleets. The second set of experiments is conducted to evaluate the computational performance under real world data sharing scale and to demonstrate the value of the proposed algorithm. The experiments are run in MATLAB 2016a on a computer with an Intel Core™ i7-6700 CPU@3.40GHz, 64-bit Windows 10 operating system. All instance data will be shared on https://github.com/BUILTNYU upon publication of this study.

4.1. Numerical verification

Three experiments are conducted here to verify the methodology and to use the numerical examples to illustrate arguments made earlier in the study. The first and second experiments are conducted on the same one-vehicle fleet instance shown in Fig. 4. At first we run the algorithm without any IO filter, and in this case compare the optimality of the algorithm as κ is incrementally increased from 2 to 10 (keeping in mind that there are 90 feasible tours for this instance). The second experiment adds the IO filter to demonstrate how the solution changes with the additional condition for checking tour feasibility. The third experiment verifies that the method can be used to protect the privacy of multi-vehicle fleet tour data.

Single vehicle fleet – no IO filter

The real tour is a single vehicle serving three passengers with the pick-up and drop-off locations shown in Fig. 4(a). The sequence, arrival time at each node, and average travel times are shown in Fig. 4(b). The vehicle is assumed to have a capacity of three people so that potentially all three passengers can be picked up before any are dropped off. Dwell time is set to zero.

The proposed algorithm in Fig. 6 is applied to the example for $\kappa = 2, \dots, 10$ and $\Delta = 0.1$ while ignoring the IO filter. The entropy value as a function of the κ is shown in Fig. 7 alongside the unconstrained upper bound and the "naïve" solution based on sorting the shortest tours and adding them incrementally. First, we note that indeed, optimality is not guaranteed when κ is small, as the iteration with $\kappa = 2$ shows the proposed algorithm underperforming the naïve solution. This is likely because there are so many feasible tours that when simply searching the most different tour from the true tour results in one that doesn't really perform well under the constrained SP2 compared with the shortest tour. However, as κ increases to 10 we see that the proposed algorithm establishes a firm gap above the naïve algorithm. We can further see that, when compared against the unconstrained upper bound, the proposed algorithm cuts the entropy from the naïve algorithm by half upon reaching $\kappa = 10$. However, the entropy assumes all tours are realistic.

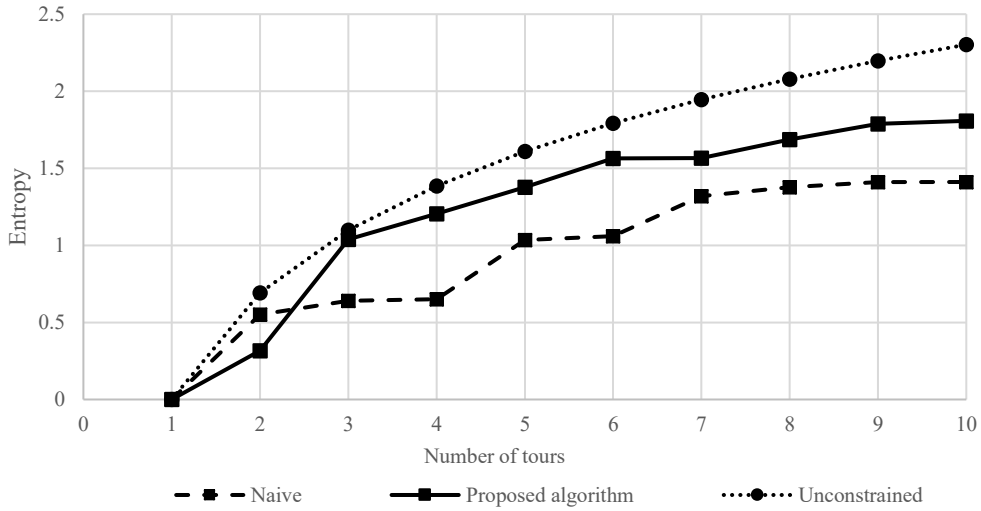


Fig. 7. Comparison of algorithms against upper bound for $\kappa = 1, \dots, 10$.

For the $\kappa = 10$ case, the ten tours found are shown in Table 2. When ignoring the IO filter, it is possible that many tours may be generated that would not be deemed "realistic" because the IO of that tour results in all zeros for the objective coefficients. In this table, 80% of the tours do not

pass that test. If we take a look at r_2, for example, going from node 1 to node 2 and then to node 4 and node 3 is highly illogical in terms of minimizing tour cost, wait time, and ride time.

Based on only the tour diffusions, the anonymity of the true tour (first one) is successfully maintained since its probability is 0.33 while the highest probability is assigned to r_5 (0.34). Adversaries would not able to identify which tour is the real one. The entropy and upper bound based on those two tours are $E^* = 0.631$ and $\hat{E}_2 = 0.693$, respectively.

Table 2 Tour-set diffusion and IO parameters without IO filter

k	tour r_k	α	β	γ	Passenger ride time			x_k
					1	2	3	
1	0-2-1-3-5-4-6-0	1.00	0.25	1.00	13.596	13.262	14.891	0.33
2	0-3-1-4-2-6-5-0	0.00	0.00	0.00	8.062	18.662	84	0.00
3	0-1-2-5-3-4-6-0	0.00	0.25	0.00	34.322	7.28	8.485	0.13
4	0-3-1-6-2-4-5-0	0.00	0.00	0.00	34.786	12.973	16.142	0.03
5	0-1-4-3-2-5-6-0	0.00	0.00	0.00	8.062	7.28	20.28	0.34
6	0-2-3-6-1-5-4-0	0.00	0.00	0.00	11.565	87.838	60.121	0.00
7	0-1-3-4-2-6-5-0	0.00	0.00	0.00	10.214	18.662	28.857	0.04
8	0-3-2-5-1-6-4-0	0.00	0.00	0.00	17.437	9.606	25.322	0.05
9	0-2-1-5-3-4-6-0	0.00	0.00	0.00	82.048	11.232	11.508	0.00
10	0-3-6-1-2-4-5-0	0.00	0.00	0.00	12.639	62.373	8.485	0.08

Single vehicle fleet – IO filter

Table 3 shows the results of the proposed algorithm with IO filter. Because of the additional feasibility condition of the IO filter, the resulting entropy is now $E^* = 1.317$ with upper bound $\hat{E}_8 = 2.079$. Even though there are only eight anonymous objects obtained out of 64 objects selected in total, all the objects are verified by the IO filter, which means they are all realistic. Based on this outcome, the standard errors are 0.105 for α and 0.112 for γ. Comparing the diffusions of the real object with IO filter to diffusions without IO filter, it is easy to conclude that the IO filter is necessary for the proposed algorithm.

Table 3 Tour-set diffusion and IO parameters with IO filter

k	tour r_k	α	β	γ	Passenger ride time			x_k
					1	2	3	
1	0-2-1-3-5-4-6-0	1.00	0.25	1.00	13.596	13.262	14.891	0.62
2	0-1-2-5-3-4-6-0	0	0.25	0	34.322	7.28	8.485	0.01
3	0-2-1-5-3-4-6-0	0.48	0.25	1.00	19.394	11.232	11.508	0.09
4	0-1-3-4-6-2-5-0	0.87	0.25	0.01	10.214	7.28	11.508	0.10
5	0-1-2-5-3-4-6-0	0.16	0.25	0.61	25.099	7.28	11.508	0.08
6	0-1-2-4-3-5-6-0	0.34	0.25	1.00	12.639	20.801	10.606	0.06
7	0-1-2-4-5-3-6-0	0.50	0.25	1.00	12.639	12.973	8.485	0.02
8	0-2-1-5-4-3-6-0	0.56	0.25	1.00	11.565	11.232	8.485	0.02
Average		0.80	0.25	0.86	15.779	12.765	12.917	
Std. error		0.105	0	0.112	2.190	0.992	0.941	

Multi-vehicle fleet

Based on the same network, we run the Fig. 6 algorithm for the multi-vehicle scenario with a maximum of 6 iterations as the stopping condition. In Fig. 8, there are two vehicles serving three passengers simultaneously. The green and blue lines represent two observed tours. Travel sequences, arrival times at each node, and real travel time of each passenger are included in Table 4. Similarly, we run the proposed algorithm with and without IO filter.

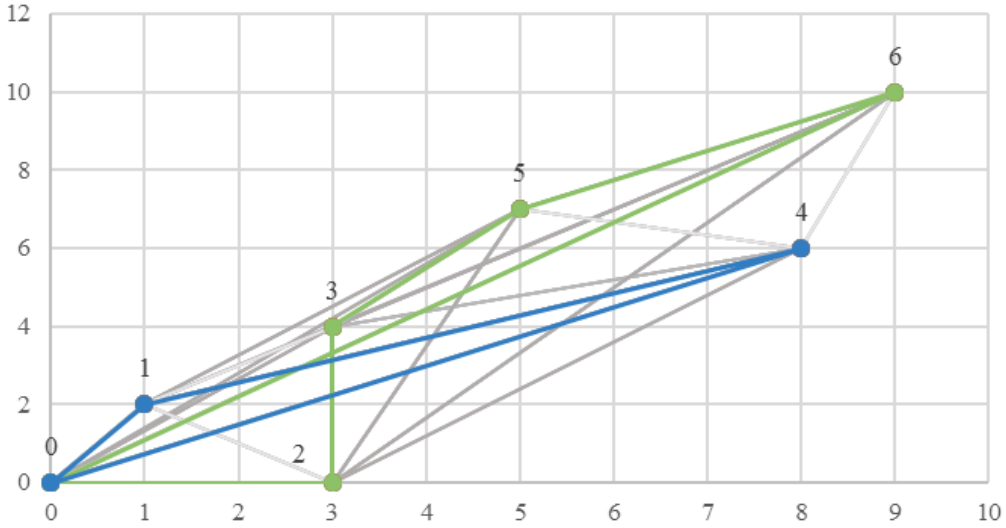


Fig. 8. Vehicle tours with a fleet of two vehicles.

Table 4 Sequence and arrival time of real tour for two-vehicle fleet example

Node	**Arrival Time**	**Node**	**Arrival Time**
0	0	0	0
1	2.236	2	3
4	10.298	3	7
0	20.298	5	10.605
		6	15.605
		0	29.059
Passenger	True travel time	Passenger	True travel time
Passenger 1 true travel time $(t_4 - t_1)$	8.062	Passenger 2 true travel time $(t_5 - t_2)$	7.605
		Passenger 3 true travel time $(t_6 - t_3)$	8.605

Table 5 Tour-set diffusion and IO parameters without IO filter

k	**tour set r_k**	α	β	γ	**Passenger ride time**			x_k
					1	2	3	
1	0-2-3-5-6-0 0-1-4-0	1	0.25	1	8.062	9.606	10.606	0.348
2	0-2-5-3-6-0 0-1-4-0	1	0.25	0.2429	8.062	7.28	8.485	0.433
3	0-3-1-6-4-0 0-2-5-0	0	0	0	17.437	7.28	16.142	0.017
4	0-1-2-4-5-0 0-3-6-0	0	0	0	12.639	984.768	8.485	0.053
5	0-3-2-6-5-0 0-1-4-0	0	0	0	8.062	18.662	17.662	0.149

From Table 5, we conclude that even if the proposed algorithm without IO generates more diffused tour sets, most of them are unrealistic. Running the algorithm with IO filter, on the other hand, generates the k-anonymous dataset more efficiently as shown in Table 6. The standard error

of γ without the IO filter is 0.27. Note that with 2 realistic tours, the unconstrained upper bound entropy is $\widehat{\mathrm{E}}_2 = 0.693$ while the solution is $\mathrm{E}^* = 0.687$.

Table 6 Tour-set diffusion and IO parameters with IO filter

k	tour set r_k	α	β	γ	Passenger ride time			x_k
					1	2	3	
1	0-2-3-5-6-0 0-1-4-0	1	0.25	1	8.062	9.606	10.606	0.794
2	0-2-5-3-6-0 0-1-4-0	1	0.25	0.2429	8.062	7.28	8.485	0.103
3	0-1-3-4-6-0 0-2-5-0	0.479	0.250	1.000	10.214	7.28	11.508	0.103
Average		0.95	0.25	0.92	8.283656	9.126844	10.48044	
Std. error		0.09	0	0.13	0.377656	0.543116	0.420871	

We note that more of the parameters exhibit a standard error with the IO filter in place. The upper bound with three tours is $\widehat{E}_3 = 1.099$ while the algorithm produces $E^* = 0.651$. The experiments demonstrate that the proposed algorithm is effective in generating tours, although adding the IO filter can lead to poor performing entropy for a given tolerance.

4.2. Case study: Long Island simulated rideshare data

This section describes a real-world case study in Long Island, New York. Since the focus of this study is on the ability to control for the privacy of a data set, as long as the data is created from a methodology that is not a direct solution of a DARP it should be sufficiently valid. The data set is obtained from dynamic routing algorithms from Ma et al. (2019). This data set is used because the outcome tours are derived in a complex manner: they are the culmination of implementing policies for dynamic dispatch, routing, idle vehicle rebalancing, and drop-offs and pickups at transit stations. In addition, the vehicles act as microtransit by providing shared rides for up to 4 passengers at a time.

The policies are operated under a simulated scenario in which the demand data is drawn from real trip data from the 2010/2011 NYMTC Regional Household Travel Survey (NYMTC, 2018). The data corresponds to trips made between 7:00 to 9:00 AM for travelers commuting to and from Long Island to New York City. The rideshare algorithm in the Ma et al. study either drops passengers off at the final destination or at a LIRR commuter rail station (or vice versa).

There are 1440 vehicle trajectories in total and distances between each two nodes are calculated using Euclidian distance. Assuming the travel speed in Long Island is 60 km/h (mostly express road speed in sub-urban area), the travel time between two nodes is calculated with Eq. (14).

$$t_{ij} = \frac{d_{ij}}{60}, \forall i, j \in N \tag{14}$$

We assume that each vehicle tour is operated independently of the other vehicle tours. For demonstration purposes, we select only thirty tours, as shown in Fig. 9, from the full tour data set to make it easier to analyze them in detail. There are three depots for the thirty tours, which are shown as yellow stars in Fig. 9 and different tours are represented in different colors. The following tests are conducted:

- An evaluation of the computation time of the algorithm in the larger scale setting;
- A sensitivity test of the privacy control under different Δ settings to demonstrate the use of the mechanism;
- A test demonstrating the effectiveness of the algorithm to adversarial attack and to user queries.

Fig. 9. Example vehicle tours in Long Island.

4.3. Computation time analysis

In this test we examine the computational performance of the algorithm. In the simulated tour data, most of tours have two or three passengers. For tours with two passengers, there are only 6 different possible tours to select, which means $\kappa \leq 6$. To keep a high computational efficiency, we define stop criteria for tour data selection: $k = 5 \; or \; |U| = 200$. Fig. 10(a) shows the breakdown of the computation time for one example tour with three passengers.

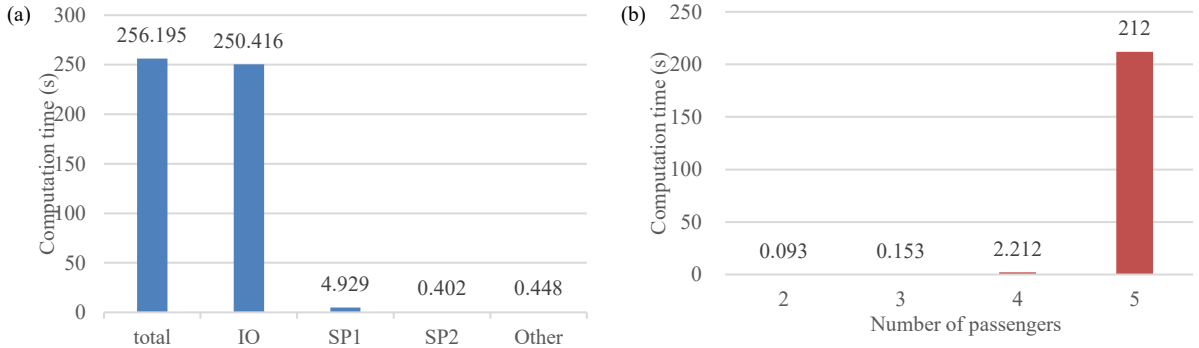


Fig. 10. (a) Computation time of an example tour data of three passengers; (b) Computation time of applying IO filter to one tour data.

In Fig. 10(a), we can see that most of computation time is spent on the IO filter. As the number of passengers increases, the computation time of applying IO filter to one tour data object increases exponentially, as Fig. 10(b) shows. The computation time for five passengers spikes up considerably, perhaps because of the implicit presence of the 4-passenger capacity.

Fig. 11 presents the relationship between the number of IO-feasible tour data objects selected among those identified in SP1. If it takes many tours to be filtered, i.e. selected with SP1, before reaching the desired κ, that is inefficient. Two trends are observed. First, most of the 30 instances can be diffused to $\kappa = 5$ IO-feasible tours within 100 identified tours. For these tours, our method works very efficiently. Second, the rest of tours are unable to reach $\kappa = 5$ IO-feasible tours even after identifying 200 tours with SP1. As shown in Fig. 11, this subset obtains two, three or four tours efficiently but becomes inefficient beyond that. We assume that for these tours, they only have a limited number of feasible tour data objects and it's necessary to set a stop criterion for the tour IO filtering in consideration of computational efficiency.

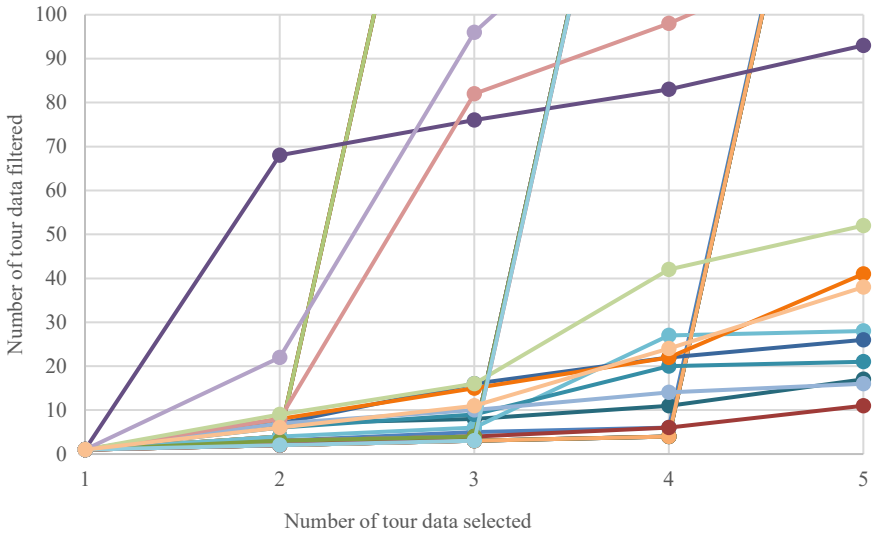


Fig. 11. Relationship between number of IO-feasible tour data objects selected from those filtered.

4.4. Results analysis

Implementing the κ-anonymous tour diffusion method on the 30 true tour data records leads to a diffused dataset. The IO-feasible 5-anonymous diffusion for one sample record (identified as 1-2-4-3-5-6) is shown in Table 7 and the diffused tours are plotted in Fig. 12. The yellow star is the depot, where vehicle starts and ends tour. The red tour is the real tour. From Table 7, we can see that as Δ changes from 0.2 to 0.3, the probabilities assigned to each synthetic tour becomes more diffused, resulting in an increase in entropy from 1.463 to 1.593.

Table 7 Example of tour diffusion with different delta

number	tours	delta=0.2 diffusion	delta=0.3 diffusion	α	β	γ
1	1-2-4-3-5-6	0.415	0.209	1	1	1
2	1-4-2-5-3-6	0.138	0.220	1	0	0.1864
3	1-4-2-3-5-6	0.217	0.220	1	0	1
4	3-6-1-4-2-5	0.137	0.220	1	0	0.167
5	1-3-2-6-5-4	0.093	0.131	0.0415	0	1

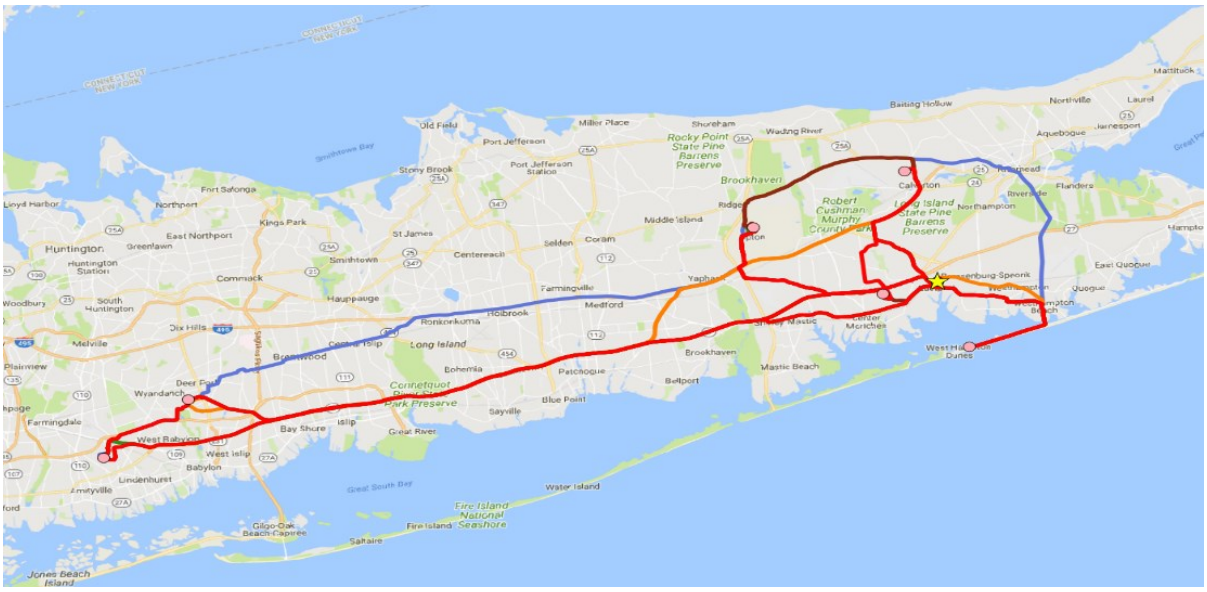


Fig. 12. An example of tour and diffusions.

To demonstrate the standard error that arises from these average values, suppose *an adversary* runs the query ten times for one of these records. Each time, they run IO on the drawn results. The result of IO inference is shown in Table 8. The standard errors over these 10 queries under different values of Δ are shown at the bottom of the table. The standard errors increase when Δ increases. This empirically demonstrates how the privacy control can increase the standard error, and how this controlled by the degree of Δ in passenger ride time accuracy. Still, this result alone does not prove there is an explicit equivalency or relationship; further research is needed to study this relationship between the entropy maximization and the IO error maximization.

Table 8 Average IO parameters of ten samples with different delta

	$\Delta = 0.2$			$\Delta = 0.3$		
	α	β	γ	α	β	γ
1	0.35	0	1	0.35	0	1
2	0.35	0	1	0.35	0	1
3	0.35	0	1	0.35	0	1
4	0.35	0	1	0.35	0	1
5	0.35	0	1	1	0	1
6	1	0	1	1	0	0.77
7	0.35	0	1	0.35	0	1
8	0.35	0	1	0.35	0	1
9	0.35	0	1	1	0	1
10	1	0	0.77	1	0	0.86
Average	0.48	0	0.98	0.61	0	0.96
Std. error	0.28	0	0.07	0.34	0	0.08

Now suppose *a user* queries the dataset to access the 30 records. The privacy control mechanism would randomly draw from the diffused tours to return one synthesized set of 30 records. Suppose the user runs the query ten times and average the passenger ride time across the 30 synthetic records in each query. The Fig. 13 presents the passenger ride time accuracy under different values of Δ. The average passenger ride times under both $\Delta = 0.2$ and $\Delta = 0.3$ are within the error threshold. Comparing ride times under $\Delta = 0.2$ with $\Delta = 0.3$, we find that ride times under $\Delta = 0.2$ are closer to the real travel time, like the 3rd object and 22nd object. Table 9 summarizes the averages over the 10 queries and shows how the standard error increases with Δ.

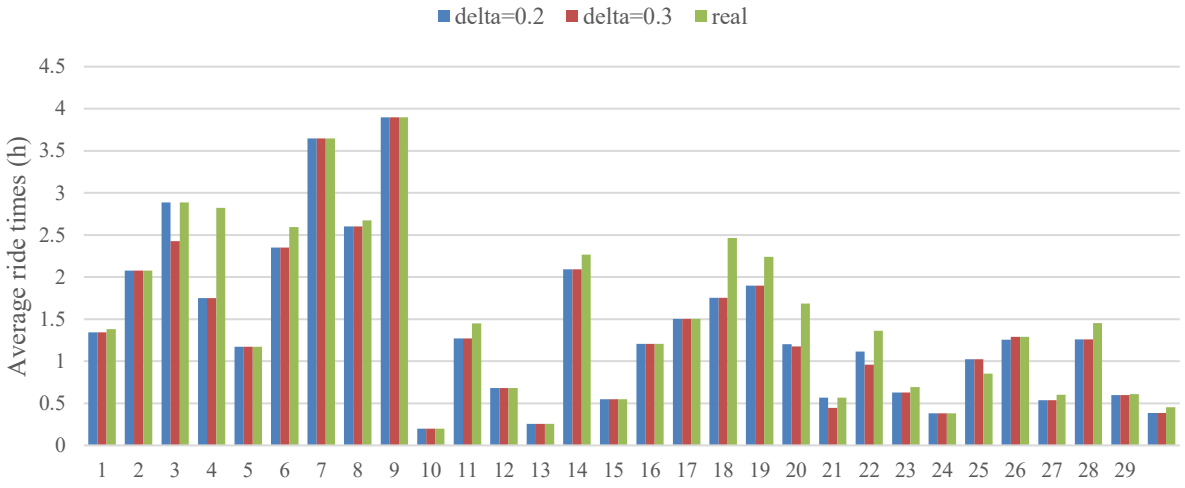


Fig. 13 Average passenger ride times of all 30 objects in one sample under different *Δ*.

Table 9 Average passenger travel times of ten samples of one object with different *Δ*

Sample	**$\Delta = 0.2$**	**$\Delta = 0.3$**
1	3.90	3.90
2	3.90	3.90
3	3.90	3.90
4	3.90	2.75
5	3.90	3.90
6	3.90	2.75
7	2.11	2.11
8	3.90	3.90
9	2.75	2.75
10	3.90	3.90
Average	3.61	3.37
Std. error	0.64	0.70

5. Conclusion and future work

Ultimately, operator data is exposed by virtue of operating in a public setting. The advantage of not making the data automatically public on an open data exchange is that building up adversarial databases take enough time and resources that an innovating company that continuously improves and modifies their algorithms should be able to nullify the value of older collected data. The key point is that the same network-oriented data set (e.g. vehicle route data) shown from different perspectives (vehicle locations, routes, pickups, etc.) can have different risks for adversarial attacks and it is a nontrivial problem to quantify these risks. Our study opens the door to studying the privacy control of network-oriented data sharing.

We propose a first κ-anonymous diffusion mechanism to control the privacy of operators' network-oriented data to address the increasingly urgent problem of data sharing between private operators and public agencies. A heuristic algorithm that is shown to be exact when applied to the full enumerated data set has several applications in reality, including idle vehicle assignment decision, transaction timestamp, vehicle tour decision, etc. To summarize, we made the following contributions to the literature:

- Proposed an algorithm to generate synthetic network data objects (tours) such that accuracy of certain desired performance measures of the data (e.g. passenger ride time) can be controlled for (generally a heuristic, but exact when applied with a full enumerated data object set);
- Proved the sublinear convergence rate of the algorithm with measurable upper bound gap;
- Numerically investigated the effectiveness of applying an IO filter to ensure only "realistic" data objects are generated, resulting in higher standard errors for reverse engineering attempts, but at a significant computational cost;
- Numerically verified the applicability of the algorithm to multi-vehicle fleet tour data (although it is shown to have more significant performance issues in identifying realistic tours with IO filter);
- Conducted a case study using realistic tour data from a Long Island travel study by Ma et al. (2018) and found:
    - A threshold in number of passengers for the tour where computational performance jumps;
    - Under certain stopping conditions the diffused data can be used to synthesize query responses that have Δ-controllable standard errors for IO-reversed parameters (for adversaries) and average passenger ride times (for users).

In the future, other applications of the κ-anonymous diffusion model should be tested and verified. For example, we can study κ-anonymous idle vehicle assignment diffusion algorithms to preserve the privacy of operators' vehicle dispatching strategies. Key points would be the design of SP2, how to select data objects to be diffused with the real data object in the direction of maximizing anonymity, and the definition of IO filter. A better search routine that more directly incorporates the IO filter as an explicit constraint or objective would likely lead to improved computational performance of the algorithm. For example, multi-armed bandit algorithms incorporate a nonlinear objective that maximizes the L_2-norm of solutions found from prior solutions (see Zhou et al., 2019), which may be a more effective formulation for SP1. Alternatively, Bell et al. (1993) proposed a column generation approach that maximize entropy; this approach might be adapted to the original formulation of Eq. (2) to implicitly enumerate consistent routes. The IO constraints, i.e. the duality conditions in Eq. (1), would be modified to

be incorporated directly into the formulation of Eq. (2). These efforts will be investigated in future research.

A prototype privacy mechanism applied to a real database of operator would be useful for practitioners. The error tolerance Δ is a required variable from public agencies to ensure the accuracy of shared data. This may constrain the efficiency of privacy preservation methods. Another idea is designing the Δ according to what kind of learning the public agencies are looking for. The level of tolerance may be subject to many factors, like size of market, the threat level over time, etc. Having designed a privacy control mechanism, we can embed this into a network design problem so that, in the same spirit of Dong et al. (2015), we can simultaneously design a network and the tolerance Δ by endogenously capturing the effect on the data quality needed for calibrating the parameters. Blockchain designs can be considered for such data exchanges as well.

In this study we chose the most complex type of network data object to test our mechanism on (NP-hard vehicle routing problems with integer programming inverse optimization) to get a sense of the computational boundaries. There are also many other network data objects that can be solved more efficiently without resorting to NP-hard problems or integer programming-based inverse optimization (e.g. shortest paths, assignment, passenger pickup/drop-off locations, fares paid, group sizes, etc.). In future research more efficient solution algorithms will be studied for this type of data object, but as other simpler data objects are explored we should likely see the computational burden taper down. For example, inverse shortest path problems can be solved very efficiently (Burton and Toint, 1992) so the hurdle that we are experiencing in the computational cost observed in Fig. 10 should subside significantly for that type of data.

Acknowledgments

This research was supported by an NSF CAREER grant, CMMI-1652735, which we gratefully acknowledge. Helpful comments from Professor Daniel Rodriguez-Roman from UPRM are also appreciated.

References

Abowd, J. M., & Lane, J. (2004). New approaches to confidentiality protection: Synthetic data, remote access and research data centers. In International Workshop on Privacy in Statistical Databases (pp. 282-289). Springer, Berlin, Heidelberg.

Ahuja, R.K. and Orlin, J.B. (2001). Inverse optimization. *Operations Research* 49(5), 771-783.

Bell, M.G.H., Lam, W. H., Ploss, G., & Inaudi, D. (1993). Stochastic user equilibrium assignment and iterative balancing. *Proc. 12th International Symposium on Transportation and Traffic Theory*. Elsevier, New York, pp.427-39.

Belletti, F., & Bayen, A. M. (2017). Privacy-preserving MaaS fleet management. *Transportation Research Part C*, in press, doi: 10.1016/j.trc.2017.08.028.

Birge, J. R., Hortaçsu, A., & Pavlin, J. M. (2017). Inverse optimization for the recovery of market structure from market outcomes: An application to the miso electricity market. *Operations Research*, 65(4), 837-855.

Bloomberg View (2017). Cities Need Data from Uber and Lyft. the Bloomberg View, July 5.

Burton, D., & Toint, P. L. (1992). On an instance of the inverse shortest paths problem. *Mathematical Programming* 53(1-3), 45-61.

Chan, T. C., Craig, T., Lee, T., & Sharpe, M. B. (2014). Generalized inverse multiobjective optimization with application to cancer therapy. *Operations Research*, 62(3), 680-695.

Chen, R., Fung, B., Desai, B.C., and Sossou, N.M. (2012). Differentially private transit data publication: a case study on the Montreal transportation system. Proc: 18th ACM SIGKDD, 213-221.
Chow, J. Y. J., & Recker, W. W. (2012). Inverse optimization with endogenous arrival time constraints to calibrate the household activity pattern problem. *Transportation Research Part B: Methodological*, *46*(3), 463-479.
Colpaert, P., Van Compernolle, M., De Vocht, L., Dimou, A., Vander Sande, M., Verborgh, R., Mechant, P. and Mannens, E. (2014). Quantifying the interoperability of open government datasets. *Computer*, *47*(10), 50-56.
Cordeau, J.F., Laporte, G. (2007). The dial-a-ride problem: models and algorithms. *Annals of Operations Research* 153(1):29–46.
Dong, R., Krichene, W., Bayen, A.M., and Sastry, S.S. (2015). Differential privacy of populations in routing games. Decision and Control (CDC), 2015 IEEE 54th Annual Conference on. IEEE.
Djavadian, S., Chow, J. Y. J. (2017). An agent-based day-to-day adjustment process for modelling 'Mobility as a Service' with a two-sided flexible transport market. *Transportation Research Part B* 104, 36-57.
Dwork, C. (2006). Differential privacy. Proc. 33rd Int. Conf. Automata Languages Program., pp. 1–12.
Dwork, C. (2008). Differential privacy: A survey of results. International Conference on Theory and Applications of Models of Computation (pp. 1-19). Springer Berlin Heidelberg.
FiveThirtyEight (2016). Uber TLC FOIL Response. https://github.com/fivethirtyeight/uber-tlc-foil-response, last accessed June 12, 2018.
Frank, M., & Wolfe, P. (1956). An algorithm for quadratic programming. *Naval Research Logistics (NRL)*, 3(1-2), 95-110.
He, Y., Chow, J.Y.J., Nourinejad, M. (2017). A privacy design problem for sharing transport service tour data, *Proc. IEEE ITS Conference*, Yokohama, Japan.
Hensher, D. A. (2017). Future bus transport contracts under a mobility as a service (MaaS) regime in the digital age: Are they likely to change?. *Transportation Research Part A: Policy and Practice*, 98, 86-96.
Jaffe, E., "Uber and Public Transit Are Trying to Get Along", Citylab, August, 3,2015.
Janssen, M., Charalabidis, Y., & Zuiderwijk, A. (2012). Benefits, adoption barriers and myths of open data and open government. *Information systems management*, *29*(4), 258-268.
Kargl, F., Friedman, A., Boreli, R. (2013). Differential privacy in intelligent transportation systems. In Proc. sixth ACM conference on Security and privacy in wireless and mobile networks (pp. 107-112). ACM.
LADOT (2018). Mobility Data Specification, https://ladot.io/wp-content/uploads/2018/12/What-is-MDS-Cities.pdf, last accessed June 11, 2019.
Le Ny, J. and Pappas, G.J. (2014). Differentially private filtering. *IEEE Transactions on Automatic Control* 59(2), 341-354.
Ma, T. Y., Rasulkhani, S., Chow, J. Y. J., Klein, S. (2019). A dynamic ridesharing dispatch and idle vehicle repositioning strategy with integrated transit transfers. *Transportation Research Part E* 128, 417-442.
NYMTC, 2018. 2010/2011 Regional Household Travel Survey, https://www.nymtc.org/DATA-AND-MODELING/Travel-Surveys/2010-11-Travel-Survey, last accessed Apr 30, 2018.
Piro, G., Cianci, I., Grieco, L. A., Boggia, G., & Camarda, P. (2014). Information centric services in smart cities. *Journal of Systems and Software*, *88*, 169-188.
Rasulkhani, S., & Chow, J. Y. (2019). Route-cost-assignment with joint user and operator behavior as a many-to-one stable matching assignment game. *Transportation Research Part B: Methodological*, *124*, 60-81.
Sadik-Khan, J. (2019). Cities need scooter data, and they need to keep it safe. *Bloomberg*, June 17.
Sankar, L., Kar, S., Tandon, R., and Poor, H.V. (2011). Competitive privacy in the smart grid: An information-theoretic approach. SmartGridComm, IEEE.
Schaffers, H., Komninos, N., Pallot, M., Trousse, B., Nilsson, M., & Oliveira, A. (2011). Smart cities and the future internet: Towards cooperation frameworks for open innovation. In *The future internet assembly* (431-446). Springer, Berlin, Heidelberg.
Shaheen,S. and Cohen, A. (2016). Innovative Mobility Carsharing Outlook. Transportation Sustainability Research Center, UC Berkeley, Berkeley, CA.
Sun, Z., Zan, B., Ban, X. J., & Gruteser, M. (2013). Privacy protection method for fine-grained urban traffic modeling using mobile sensors. *Transportation Research Part B* 56, 50-69.
Sweeney, L. (2002). k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05), 557-570.
Tenorio, L. (2001). Statistical regularization of inverse problems. *SIAM Review*, 43(2), 347-366.
Trotter, J. K. (2014). Public NYC Taxicab Database Lets You See How Celebrities Tip. http://gawker.com/the-public-nyc-taxicab-database-that-accidentally-track-1646724546, last accessed June 13, 2018.

Tsai, Y.C., Wang, S.L., Kao, H.Y., and Hong, T.P. (2015). Edge types vs privacy in K-anonymization of shortest paths. *Applied Soft Computing* 31: 348-359.

Wang, L. (2009). Cutting plane algorithms for the inverse mixed integer linear programming problem. *Operations Research Letters*, 114-116.

Wang, S. L., Tsai, Y. C., Hong, T. P., & Kao, H. Y. (2017). k^--anonymization of multiple shortest paths. *Soft Computing*, 21(15), 4215-4226.

Wilson, A. G. (1967). A statistical theory of spatial distribution models. *Transportation Research* 1, 253-269.

Xu, S. J., Nourinejad, M., Lai, X., Chow, J.Y.J. (2018). Network learning via multi-agent inverse transportation problems, *Transportation Science* 52(6), 1347-1364.

You, S. I., Chow, J. Y. J., & Ritchie, S. G. (2016). Inverse vehicle routing for activity-based urban freight forecast modeling and city logistics. *Transportmetrica A: Transport Science*, *12*(7), 650-673.

Yuan, C., Thai, J., & Bayen, A. M. (2016). Zubers against zlyfts apocalypse: An analysis framework for dos attacks on mobility-as-a-service systems. In Proceedings of the 7th International Conference on Cyber-Physical Systems (p. 24). IEEE Press.

Zhou, J., Lai, X., & J. Y. J. Chow (2019). Multi-armed bandit on-time arrival algorithms for sequential reliable route selection under uncertainty. *Transportation Research Record*, in press, doi: 10.1177/0361198119850457.