Constrained-Based Differential Privacy for Mobility Services

Ferdinando Fioretto
University of Michigan
Ann Arbor, MI, USA
fioretto@umich.edu

Chansoo Lee
University of Michigan
Ann Arbor, MI, USA
chansool@umich.edu

Pascal Van Hentenryck
University of Michigan
Ann Arbor, MI, USA
pvanhent@umich.edu

ABSTRACT

Ubiquitous mobile and wireless communication systems have the potential to revolutionize transportation systems, making accurate mobility traces and activity-based patterns available to optimize the design and operations of mobility systems. However, these rich data sets also pose significant privacy risks, potentially revealing highly sensitive information about individual agents.

This paper studies how to use *differential privacy* to release mobility data for transportation applications. It shows that existing approaches do not provide the desired fidelity for practical uses. To remedy this limitation, the paper proposes the idea of *Constraint-Based Differential Privacy* (CBDP) that casts the production of a private data set as an optimization problem that redistributes the noise introduced by a randomized mechanism to satisfy fundamental constraints of the original data set.

The CBDP has strong theoretical guarantees: It is a constant factor away from optimality and when the constraints capture categorical features, it runs in polynomial time. Experimental results show that CBDP ensures that a city-level multi-modal transit system has similar performance measures when designed and optimized over the real and private data sets and improves state-of-art privacy methods by an order of magnitude.

KEYWORDS

Differential Privacy; Mobility; Transportation;

ACM Reference Format:
Ferdinando Fioretto, Chansoo Lee, and Pascal Van Hentenryck. 2018. Constrained-Based Differential Privacy for Mobility Services. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 9 pages.

1 INTRODUCTION

The availability of mobility traces and activity-based patterns has the potential to revolutionize the design and operations of transportation systems. For instance, shared-ride services such as *UberPool* and *Via* reroute driver paths in real time to optimize vehicle capacity, bike-sharing programs such as *CitiBike* in New York rebalance their fleets from popular destinations to popular origins, and *CityMapper* operates a private night bus line in London based on their analysis of the mobility data collected from user data.

However, the release of mobility data poses significant risks, as they contain highly sensitive information about individual agents,

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

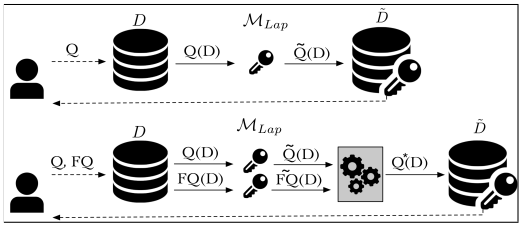

Figure 1: Comparison of the direct and CBDP approaches. The direct approach (top) uses the output $\tilde{Q}(D)$ from the Laplace mechanism $\mathcal{M}_{\text{Lap}}$ to construct the differentially-private data set $\tilde{D}$. CBDP (bottom) issues additional queries and uses an optimization model to redistribute the noise.

Indeed, De Montjoye et al. [4] have shown that only four spatio-temporal points are sufficient to uniquely identify 95% of the individuals within the spatial resolution of an antenna. It is thus not surprising that the release of private mobility data has attracted significant attention in recent years [1, 10, 15].

In the last decade, *differential privacy* [6] has emerged as the de-facto standard for protecting the privacy of individual agents. Informally speaking, differential privacy bounds the *harm* caused by the participation of an agent to a data set. To protect individual agents, differential privacy uses randomized mechanisms that inject noise in the data. The magnitude of the random noise impacts the utility of the data set, creating a tradeoff between utility and privacy [12, 17, 21].

This paper is motivated by the release of private data sets for the design and operations of On-Demand Multimodal Transit Systems (ODMTS), where high-frequency buses or light-rail lines are combined with on-demand shuttles to address both congestion and the first/last mile problem [19]. It considers a case study where a city desires to release temporal Origin-Destination (O-D) pairs of its population so that third-parties can propose configurations of the new mobility services. This setting is ideal for the application of differential privacy. Unfortunately, experimental results on the city of Ann Arbor in Michigan shows that the optimal design of the ODMTS over the private data set is about 150% more expensive than over the original data set. The noise added by the standard differential privacy mechanism fails to preserve the combinatorial structure of the original data.

To remedy this limitation, this paper proposes the framework of *Constrained-Based Differential Privacy* (CBDP), whose key idea is to use optimization technology to redistribute the noise introduced by standard differential privacy mechanisms and preserve salient features of the application. In a first step, CBDP applies a standard differential privacy mechanism on two types of queries: the *traditional queries* needed to construct the private database and

a collection of *feature queries* that capture important *constraints* of the targeted application. Since differential privacy mechanisms introduce noise independently on each query, the resulting query outputs, and the private data set, are inconsistent. The second step of CBDP uses an optimization model to redistribute the noise in order to restore consistency and ensure that the noise stays as close as possible to the one in the standard mechanism. Figure 1 highlights the difference between the direct (top row) and the CBDP (bottom row) approaches.

The paper shows that the CBDP has strong theoretical properties: it achieves ϵ-differential privacy, it ensures query consistency, and it is a constant factor away from optimality. When the constraints capture categorical features, the CBDP mechanism also runs in polynomial time. Finally, experimental results shows that the CBDP mechanism is also practical: On the design of an ODMTS for the city of Ann Arbor, MI, it improves the accuracy existing approaches by an order of magnitude and results in negligible utility losses.

2 DIFFERENTIAL PRIVACY

This section summarizes key results in Differential Privacy (DP) [8]. A randomized mechanism $\mathcal{M} : \mathscr{D} \to \mathscr{R}$ with domain $\mathscr{D}$ and range $\mathscr{R}$ is ϵ-differentially private if, for any $\mathcal{S} \subseteq \mathscr{R}$ and any two inputs $D_1, D_2 \in \mathscr{D}$ differing in at most one data item[1] (written $||D_1 - D_2||_1 \leq 1$),

$$Pr[\mathcal{M}(D_1) \in \mathcal{S}] \leq \exp(\epsilon) Pr[\mathcal{M}(D_2) \in \mathcal{S}], \tag{1}$$

where the probability is calculated over the coin tosses of $\mathcal{M}$. The parameter ϵ is the *privacy budget* of the mechanism.

Differential privacy satisfies several important properties, including *composability* and *immunity to post-processing*. Composability ensures that a combination of differentially private mechanisms preserves differential privacy.

Theorem 2.1 (Composition Theorem). *Let $M_i : \mathscr{D} \to \mathscr{R}_i$ be an ϵ_i-differentially private mechanism $(1 \leq i \leq k)$. Their composition $\mathcal{M}(D) = (\mathcal{M}_i(D), \ldots, \mathcal{M}_k(D))$ is $(\sum_{i=1}^{k} \epsilon_i)$-differentially private.*

The immunity to post-processing ensures that applying arbitrarily many functions to the output of a differentially private-mechanism preserves its privacy guarantees.

Theorem 2.2 (Post-Processing Immunity). *Let $\mathcal{M} : \mathscr{D} \to \mathscr{R}$ be an ϵ-differentially private mechanism and $g : \mathscr{R} \to \mathscr{R}'$ be an (arbitrary) mapping. The mechanism $g \circ \mathcal{M}$ is ϵ-differentially private.*

In private data analysis settings, agents interact with the data set by issuing queries. A (numeric) *query* is a function from a data set $D \in \mathscr{D}$ to a result set $\mathscr{R} \subseteq \mathbb{R}^n$. The *sensitivity* of a query Q, denoted by Δ_Q, is defined as

$$\Delta_Q = \max_{\substack{D_1, D_2 \text{ s.t.} \\ ||D_1 - D_2||_1 \leq 1}} ||Q(D_1) - Q(D_2)||_1. \tag{2}$$

Counting queries are particularly useful and return the number of data points satisfying a predicate. They have sensitivity 1.

Adding Laplacian noise when answering numeric queries produces a differentially private mechanism [6].

[1] I.e., $|(D_1 - D_2) \cup (D_2 - D_1)| = 1$

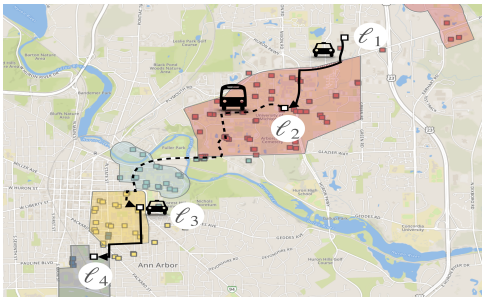

Figure 2: A Multimodal Transit System. Squares denote boarding/alighting stops. Colors highlight different service zones. The line segment illustrates a multi-modal three-legs route (shuttle-bus-shuttle).

Theorem 2.3 (Laplace Mechanism). *Let $Q : \mathscr{D} \to \mathscr{R}$ be a numerical query. The Laplace mechanism, defined as $\mathcal{M}_{Lap}(D; Q, \epsilon) = Q(D) + z$ where $z \in \mathscr{R}$ is a vector of i.i.d. samples drawn from the Laplace distribution with scaling factor Δ_f / ϵ, is ϵ-differential private.*

3 ON-DEMAND MULTIMODAL TRANSIT

On-Demand multimodal transit systems (ODMTS) jointly address congestion and the first/last mile problem that plagues many transit systems. They combine high-frequency buses (or light-rail) between hubs for high-density corridors, with on-demand shuttles to bring riders from their origins to the hubs and from the hubs to their destinations. On-demand shuttles also perform direct trips between origins and destinations. Bus routes are fixed, while the shuttles are dispatched and routed dynamically, maximizing ride sharing while preserving short waiting and transit times for passengers. Riders are picked up and dropped off at virtual vehicle stops (also called *locations* for simplicity), which include the hubs. Figure 2 illustrates a trip with three legs in an ODMTS consisting a shuttle leg from the trip origin (location ℓ_1) to a hub (location ℓ_2), followed by a bus leg to another hub (location ℓ_3) and a final shuttle leg to the final destination (location ℓ_4).

The design of an ODMTS aims at minimizing the total cost of the system (investment and operating costs), while keeping the average transit time within a desired interval. Maheo et al. [19] proposed an advanced Benders decomposition algorithm to design an optimal ODMTS from a data set of temporal O-D pairs. When scaling to large cities or significant riderships, it is also desirable to partition the transit region into zones, which are operated and optimized independently. In particular, each zone are allocated a number of shuttles that serve the requests within the zone and from the zone to a hub (and vice-versa). In Figure 2, the zones are depicted by the color of the stops. The design of an ODMTS must then optimally size the fleet of each zone and determine a cost-optimal rostering of the drivers that obeys federal and state regulations. Finally, during operations, real-time algorithms for shuttle dispatching, routing, and ride-sharing aim at minimizing the average transit times of riders, while satisfying convenience constraints.

For concreteness, the paper focuses on the following case study. A transit agency has designed an ODMTS, including the bus network and the shuttle zones, and would like to engage a mobility

solution provider for servicing the shuttle component. To help the procurement process, the transit agency would like to release a private data set so that mobility solution providers can size their fleets properly to minimize cost and satisfy the selected performance criteria for waiting and transit times. Mobile solution providers will then use their optimization algorithms for fleet sizing and real-time operations, denoted by $\mathcal{F}$ and O respectively, in order to price their services accurately. This case study is representative of many procurement and competition settings.

4 NOTATIONS AND SETTINGS

Let $\mathbb{N}_0$ be the set of non-negative integers and $\mathscr{U}$ the *data universe*, which we assume to be a finite set. A data set D is a multi-set of elements in $\mathscr{U}$ or, equivalently, a vector in $\mathbb{N}_0^{|\mathscr{U}|}$. The data universe for our mobility data set is defined as $\mathscr{U} = \mathbf{L}\times\mathbf{L}\times\mathbf{T}$, where $\mathbf{L}$ is a set of locations (virtual vehicle stops) and $\mathbf{T}$ is a partition of a 24-hour day in time periods. The *time resolution* determines the cardinality of $\mathbf{T}$. Elements in $\mathscr{U}$ are called *trip types* and are denoted by $\langle o, d, t\rangle$, where $o, d \in \mathbf{L}$ are the origin and destination of the request and $t \in \mathbf{T}$ represents its time period. Our mobility data set is a multiset of trip types or, equivalently, a vector of type $\mathbb{N}_0^{|\mathbf{L}^2\mathbf{T}|}$.

This paper studies the release of a private version of a data set, which is then used in a procurement or competition context. The original data set can be obtained by numeric queries on every $u \in \mathscr{U}$. Similarly, a private data set can be produced via a mapping from noisy outputs $\tilde{c}_u$ to these numeric queries. As a result, this paper primarily focuses on differential-privacy mechanisms for these numeric queries.

5 RELATED WORK

As sketched in the introduction, the first step of CBDP uses a sequence of correlated (batched) queries. Research on answering batched queries has focused primarily at reducing the sensitivity of correlated queries. Xiao et al. [24] applies a wavelet transformation to the data set frequency to generate a wavelet coefficient resulting in reduced noise variance per query. Li et al. [16] proposed a *matrix* mechanism to answer linear combinations of queries. Huang et al. [13] applies a transformation to the batch of queries to output a set of orthogonal queries to reduce correlation between them. The literature on *k-way marginals* [5, 7, 22] focuses on improving the time complexity or accuracy when the data universe is so large that the full marginal is unnecessary and infeasible. *In contrast, CBDP releases the full private data set, imposes constraints to ensure its consistency, and optimization to guarantee its accuracy.*

The closest related work is the Hierarchical (H) mechanism of Hay et al. [11] and its extensions [2, 21]. It uses a post-processing step that enforces additive constraints based on a tree structure of the data universe. The mechanism creates a balanced tree where the leaves are singleton sets of the universe $\mathscr{U}$, each internal node is the union of its children, and the root is the entire universe. For a tree of height h, H first uses the Laplace mechanism for answering a counting query for each node of the tree (except the root). Each such query has a privacy parameter ϵ/h, so that the step is ϵ-differentially private. The second step of H takes the resulting tree $T_0 \in \mathbb{R}^{|\mathscr{U}|}$ and applies a variance reduction process to obtain a tree T_1. The final step in H finds a tree T in which a parent value is equal to the sum of values of its children and the value $\|T - T_1\|_2^2$ is minimized. Thanks to the tree structure and simplicity of the objective, there is a simple closed form formula to compute T. The leaves can then be used to reconstruct a data set. Although the original hierarchical mechanism only considers balanced trees and is agnostic to the data semantics, it can be altered to use features for branching and is thus directly applicable to the setting of this paper. A detailed comparison of CBDP and H is presented in Section 11.

6 THE DIFFERENTIAL PRIVACY CHALLENGE

It is useful to highlight the challenges raised by the case study. For the OBMTS of Ann Arbor, the Laplace mechanism on the batched queries outputs a real vector with many negative numbers (which are obviously semantically meaningless). A post-processing step that rounds each count to its closest non-negative number induces a significant bias towards positive noise values. This problem is exacerbated by the sparsity of our query results. For the considered OBMTS, the original data set contains 37,714 trips. The Laplace mechanism for $\epsilon = 1$ returns on average of 261,032 trips (averaged over 50 independent runs), which is about a 7 fold increase. For $\epsilon = 0.01$, the Laplace mechanism generates 3,312,350 trips just between 8am-10am compared to 4878 actual trips. As a result, a fleet-sizing optimizer will significantly overestimate the number of shuttles required and the overall operating cost of the ODMTS, making the release of the private data set useless.

To understand these results, it is useful to observe that, for a resolution of 30 minutes, there are 67600 possible trip types within the two-hour period 8am–10am. However, only 591 of them have non-zero counts. This pathological behavior of the Laplace mechanism on sparse counting queries is a well-known critical issue when applying differential privacy in practice [3, 9, 18, 23].

7 THE CBDP MECHANISM

To remedy this important issue, this paper introduces the idea of *Constraint-Based Differential Privacy* (CBDP). It is especially suitable for complex optimization tasks whose outcomes, crucially, but indirectly, depend on structural properties of the data. CBDP uses the concept of *features* to capture semantic properties of the application and queries these features in addition to the orginal data set.

Features and Feature Queries. A central element of CBDP is the concept of feature. A (categorical) *feature* is a partition of the universe $\mathscr{U}$ and the size of the feature is the number of elements in the partition. A feature $\mathbf{F}'$ is a *sub-feature* of $\mathbf{F}$, denoted by $\mathbf{F}' \prec \mathbf{F}$, if $\mathbf{F}'$ is obtained by sub-partitioning $\mathbf{F}$. The *feature query* on data set $D \in \mathscr{D}$ associated with feature $\mathbf{F} = \{\mathbf{d}_1, \ldots, \mathbf{d}_n\}$ is denoted by $Q_{\mathbf{F}}(D)$: It returns an n-dimensional count vector $(c_1, \ldots, c_n)$ where each c_i is the number of data points in D that belong to $\mathbf{d}_i$.

The Input. The mechanisms studied in this paper take as input the data set D and a collection of features $\mathscr{F} = \{\mathbf{F}_1, \ldots, \mathbf{F}_k\}$ where $\mathbf{F}_i = \{\mathbf{d}_{i1}, \ldots, \mathbf{d}_{in_i}\}$ for $(i = 1, \ldots, k)$. For simplicity, the first feature always partitions the universe into singletons, i.e., $\mathbf{F}_1 = \{\{u\} : u \in \mathscr{U}\}$, since these are the counts used to construct the private data set. The paper also assumes that the second feature contains the universe as a single element, i.e., $\mathbf{F}_2 = \{\mathscr{U}\}$. Hence, query $Q_{\mathbf{F}_1}(D)$

$$\textbf{minimize: } \|\mathbf{x}-\tilde{\mathbf{c}}\|^2_{2,\mathbf{w}} = \sum_{i=1}^{k} \frac{1}{n_i} \sum_{j=1}^{n_i} (x_{ij}-\tilde{c}_{ij})^2 \quad \text{(O1)}$$

subject to:

$$\forall i', i : \mathbf{F}_{i'} \prec \mathbf{F}_i, j \in [n_i] : x_{ij} = \sum_{l:\mathbf{d}_{i'l} \subseteq \mathbf{d}_{ij}} x_{i'l} \quad \text{(O2)}$$

$$\forall i, j : x_{ij} \geq 0. \quad \text{(O3)}$$

Figure 3: The CBDP Post-Processing Step.

reports the counts of the original data set D, while query $Q_{\mathbf{F}_2}(D)$ reports the total number of elements in the data set. When viewed as queries, the inputs to the mechanism can be represented as a set of counts $Q_{\mathbf{F}_i} = c_i = (c_{i1}, \ldots, c_{in_i})$ $(1 \leq i \leq k)$ or, more concisely, as $\mathbf{c} = (c_{11}, \ldots, c_{kn_k})$. Finally, for simplicity, the mechanisms assume that the partial ordering $\prec$ of features is given.

The CBDP Mechanism. CBDP first applies the Laplace mechanism with privacy parameter $\frac{\epsilon}{k}$ to each feature query, i.e.,

$$\mathcal{M}_{\text{Lap}}(D; Q_{\mathbf{F}_i}, \epsilon/k) = \tilde{c}_i = (\tilde{c}_{i1}, \ldots, \tilde{c}_{in_i}) \in \mathbb{R}^{n_i} \quad (1 \leq i \leq k).$$

The resulting counts $\tilde{\mathbf{c}} = (\tilde{c}_{11}, \ldots, \tilde{c}_{kn_k})$ are then post-processed by the optimization algorithm depicted in Figure 3 to obtain the counts $\mathbf{x}^* = (x^*_{11} \ldots, x^*_{kn_k})$. Finally, the CBDP mechanism outputs a data set $\tilde{D}$ such that $Q_{\mathbf{F}_1}(\tilde{D})$ returns a rounded version of $\mathbf{x}^*$. This is trivially achieved as described earlier, since our universe is finite.

The essence of the CBDP mechanism is obviously the optimization model. Its decision variables are the postprocessed counts $\mathbf{x} = (x_{11} \ldots, x_{kn_k})$, and $\mathbf{w} = (w_1, \ldots, w_k) \in (0,1]^k$ is a vector of reals representing weights for the terms of the objective function. The objective minimizes the squared weighted L_2-Norm of $\mathbf{x} - \tilde{\mathbf{c}}$, where the weight w_i of element $x_{ij} - \tilde{c}_{ij}$ is $\frac{1}{n_i}$. The optimization is subject to a set of *consistency constraints* among comparable features and non-negativity constraints on the variables. For each pair of features $(\mathbf{F}_{i'}, \mathbf{F}_i)$ with $\mathbf{F}_{i'} \prec \mathbf{F}_i$, constraint O2 selects an element $\mathbf{d}_{ij} \in \mathbf{F}_i$ and all its subsets $\mathbf{d}_{i'l} \in \mathbf{F}_{i'}$ and imposes the constraint

$$x_{ij} = \sum_{l:\mathbf{d}_{i'l} \subseteq \mathbf{d}_{ij}} x_{i'l}$$

which ensures that the postprocessed count x_{ij} is consistent with the sum of the postprocessed counts of its partition in $\mathbf{F}_{i'}$. By definition of sub-features, there exists a set of elements in $\mathbf{F}_{i'}$ whose union is equal to $\mathbf{d}_{ij}$. Note that the Laplacian counts do not generally satisfy these constraints.

Intuitively, the CBDP mechanism can be thought as redistributing the noise introduced by the Laplace mechanism to obtain a consistent data set. The post-processing step searches for a solution that satisfies all the feature constraints and is as close as possible to the Laplacian counts. A feasible solution always exists, since the real counts $\mathbf{c}$ satisfy all constraints. Observe that, when only $\mathbf{F}_1$ and $\mathbf{F}_2$ are used then the optimization model enforces the constraints

$$\sum_{j=1}^{|\mathscr{U}|} x_{1j} = x_{21}$$

and minimizes

$$(x_{21} - \tilde{c}_{21})^2 + \frac{1}{|\mathscr{U}|} \sum_{j=1}^{|\mathscr{U}|} (x_{1j} - \tilde{c}_{1j})^2$$

Together they impose a strong relationship between the post-processed individual counts and the post-processed total count, which aims at preserving the number of elements in the data set and the sparsity of the data set,

The CBDP mechanism exploits three insights. First, by using a weighted L_2-norm, CBDP ensures that the sums of the terms for each partition are of the same order of magnitude. This is natural, since they all partition the entire universe. Second, observe that the features are *not* hierachical: They can capture fundamentally different aspects of the problem structure. Finally, the non-negativity constraints ensure that only non-nengative post-processed counts are generated, contrary to the Laplacian mechanism.

8 THEORETICAL PROPERTIES OF CBDP

This section presents some theoretical properties of the CBDP.

Theorem 8.1. *CBDP achieves ϵ-differential privacy.*

Proof. Since each feature partitions the universe, each feature query is a counting query with sensitivity 1. Thus each $\tilde{c}_{ij}$ obtained from the Laplace mechanism is ϵ/k-differentially-private by Theorem 2.3. The combination of these results $(\tilde{c}_{11}, \ldots, \tilde{c}_{kn_k})$ is ϵ-differentially-private by Theorem 2.1. The result follows from post-processing immunity (Theorem 2.2). □

Observe that the mechanisms considered in this paper all operate over the universe, (e.g., the set of trip types of the form $\langle o, d, t \rangle$ in the case study). This is the case for instance of the Laplace mehanisms which runs in polynomial time in the size of the universe. The next theoretical result characterizes the complexity of the post-processing step and hence the complexity of the CBDP mechanism. Recall that a δ-solution to an optimization problem is a solution whose objective value is within distance δ of the optimum.

Theorem 8.2. *A δ-solution to the optimization to the optimization model in Figure 3 can be obtained in time polynomial in the size of the universe, the number of features, and $\frac{1}{\delta}$.*

Proof. First observe that the number of variables and constraints in the optimization model are bounded by a polynomial in the size of the universe and the number of features. Indeed, since the features are partitions, every set $\mathbf{d}_{i'l}$ in Constraint (O2) is a subset of exactly one $\mathbf{d}_{ij}$. The result then follows from the fact that the optimization model is convex, which implies that a δ-solution can be found in time polynomial in the size of the universe, the number of features, and $\frac{1}{\delta}$ [20]. □

The final theoretical results bounds the accuracy of the CBDP mechanism in terms of the accuracy of the Laplace mechanism.

Theorem 8.3. *The optimal solution to the optimization model in Figure 3 satisfies*

$$\|\mathbf{x}^* - \mathbf{c}\|_{2,\mathbf{w}} \leq 2\|\tilde{\mathbf{c}} - \mathbf{c}\|_{2,\mathbf{w}}.$$

Proof.

$$\|\mathbf{x}^* - \mathbf{c}\|_{2,w} \leq \|\mathbf{x}^* - \tilde{\mathbf{c}}\|_{2,w} + \|\tilde{\mathbf{c}} - \mathbf{c}\|_{2,w} \quad (3)$$

$$\leq 2\|\mathbf{c} - \tilde{\mathbf{c}}\|_{2,w}. \quad (4)$$

where the first inequality follows from the triangle inequality on weighted L_2-norms and the second inequality follows from

$$\|\mathbf{x}^* - \tilde{\mathbf{c}}\|_{2,w} \leq \|\mathbf{c} - \tilde{\mathbf{c}}\|_{2,w}$$

by optimality of $\mathbf{x}^*$ and the fact that $\mathbf{c}$ is a feasible solution to constraints (O2) and (O3). □

The following corollary follows from the optimality of the Laplace mechanism [14].

Corollary 8.4. *The CBDP mechanism is at most a factor 2 away from optimality.*

9 APPLICATION TO THE CASE STUDY

This section applies the CBDP mechanism to the case study in mobility. The CBDP mechanism uses the following features $\{\mathbf{F}_1, \ldots, \mathbf{F}_5\}$:

(1) $\mathbf{F}_1 = \{\{l\} : l \in \mathscr{U}\}$ is the partition into singleton sets.
(2) $\mathbf{F}_2 = \{\mathscr{U}\}$ captures the total number of trips, a key feature for the fleet sizing of ODMTS.
(3) $\mathbf{F}_3$ partitions the universe by trips occurring within a given time period t.
(4) $\mathbf{F}_4$ partitions the universe by the zones of the O-D pair (e.g., from Zone 1 to Zone 3). The amount of inter-zone mobility demands is also a key feature of ODMTS, especially for high fidelity in transit times. The set of all zones is denoted by $\mathbf{Z}$.
(5) $\mathbf{F}_5$ partitions the universe by the transportation mode m (e.g., shuttle-bus-shuttle, shuttle-bus, bus, etc.). Transportation modes have a significant effect on fleet sizing and the overall cost of the ODMTS. The set of all transportation modes is denoted by $\mathbf{M}$.

In lieu of numeric subscripts, the presentation uses the following notations for more clarity; $\tilde{c}_D, \tilde{c}_t, \tilde{c}_{lt}, \tilde{c}_{zt}$, and $\tilde{c}_{mt}$, respectively, denote the total noisy number of trips ($\mathbf{F}_2$), the noisy number of trips grouped by time intervals t ($\mathbf{F}_3$), the noisy number of trips for an O-D pair l and time t ($\mathbf{F}_1$), the noisy number of a trips for a zonal O-D pair z and time t ($\mathbf{F}_4$), and the noisy number of trips for a given transportation mode and time t ($\mathbf{F}_5$). The optimization model associates variables x_D, x_t, x_{lt}, x_{zt}, and x_{mt} with each of these noisy counts.

The optimization model, which is an instantiation of the general mechanism, is given in Figure 4. The constraints are driven from the partial ordering $\mathbf{F}_1 \prec \mathbf{F}_4 \prec \mathbf{F}_3 \prec \mathbf{F}_2$ and $\mathbf{F}_1 \prec \mathbf{F}_5 \prec \mathbf{F}_3 \prec \mathbf{F}_2$.

minimize:

$$(x_D - \tilde{c}_D)^2 + \frac{1}{|\mathbf{T}|}\sum_{t\in\mathbf{T}}(x_t - \tilde{c}_t)^2 \quad (5)$$

$$+ \frac{1}{|\mathbf{T}\times\mathbf{L}^2|}\sum_{l\in\mathbf{L}^2, t\in\mathbf{T}}(x_{lt} - \tilde{c}_{lt})^2 \quad (6)$$

$$+ \frac{1}{|\mathbf{T}\times\mathbf{Z}^2|}\sum_{z\in\mathbf{Z}^2, t\in\mathbf{T}}(x_{zt} - \tilde{c}_{zt})^2 \quad (7)$$

$$+ \frac{1}{|\mathbf{T}\times\mathbf{M}|}\sum_{m\in\mathbf{M}, t\in\mathbf{T}}(x_{mt} - \tilde{c}_{mt})^2 \quad (8)$$

subject to:

$$x_D = \sum_{(l,t)\in\mathbf{L}^2\times\mathbf{T}} x_{lt} \quad (9)$$

$$x_D = \sum_{t\in\mathbf{T}} x_t \quad (10)$$

$$x_t = \sum_{\{(l,t)\}\in\mathbf{d}_{3lt}} x_{lt} \quad (\forall t \in \mathbf{T}) \quad (11)$$

$$x_D = \sum_{(z,t)\in\mathbf{Z}^2\times\mathbf{T}} x_{zt} \quad (12)$$

$$x_t = \sum_{z\in\mathbf{Z}^2} x_{zt} \quad (\forall t \in \mathbf{T}) \quad (13)$$

$$x_{zt} = \sum_{(l,t)\in\mathbf{d}_{4zt}} x_{lt} \quad (\forall z \in \mathbf{Z}^2, t \in \mathbf{T}) \quad (14)$$

$$x_D = \sum_{(m,t)\in\mathbf{M}\times\mathbf{T}} x_{mt} \quad (15)$$

$$x_t = \sum_{m\in\mathbf{M}} x_{mt} \quad (\forall t \in \mathbf{T}) \quad (16)$$

$$x_{mt} = \sum_{(l,t)\in\mathbf{d}_{5mt}} x_{lt} \quad (\forall z \in \mathbf{M}, t \in \mathbf{T}) \quad (17)$$

$$x_D, x_t, x_{lt}, x_{zt}, x_{mt} \geq 0 \quad (\forall t \in \mathbf{T}, l \in \mathbf{L}^2, z \in \mathbf{Z}^2, m \in \mathbf{M}) \quad (18)$$

Figure 4: The CBDP Optimization Step for the ODMTS.

10 EXPERIMENTAL RESULTS

This section presents an evaluation of the CBDP mechanism on the case study. It first presents the experimental setup and compares the CBDP mechanism with the Laplace and hierarchical mechanisms.

Data sets and Experimental Setup. The experimental results concern an on-demand multi-modal transportation system for the city of Ann Arbor in Michigan. The city has 160 locations that are allotted as stops for both buses and small shuttles, which makes a total of 16,900 possible O-D pairs. Buses have a capacity of 100 passengers, while shuttles have a capacity of 8 passengers. The experiments use multiple data sets collected during a *weekday* and a *weekend* capturing user mobility activities. The data is collected through a collaboration with the urban transit system of the city under an IRB. The baseline $\mathcal{M}_{\text{Lap}}$ is the Laplace mechanism with a post-processing that rounds the results to the nearest non-negative integer. The mechanisms are implemented in *Python 3* and *Gurobi 6.0*. The fleet sizing and operational optimization algorithms $\mathcal{F}$ and $\mathcal{O}$ are implemented in *C++* with some scripting code in *Python 3*. The experiments are executed on an Intel(R) Xeon(R) E3-1240 3.50GHz with 4GB of RAM. The optimization algorithm $\mathcal{F}$ is allocated 24 hours for finding a fleet sizing that satisfies the desired expected waiting time which must be no more than 180 seconds. The resolution of the time periods is set to 30 minutes, there are 4 different mobility zones, and 12 feasible combinaitons of transportation modes.

The CBDP Mechanism. Three variants of the CBDP mechanisms were implemented:

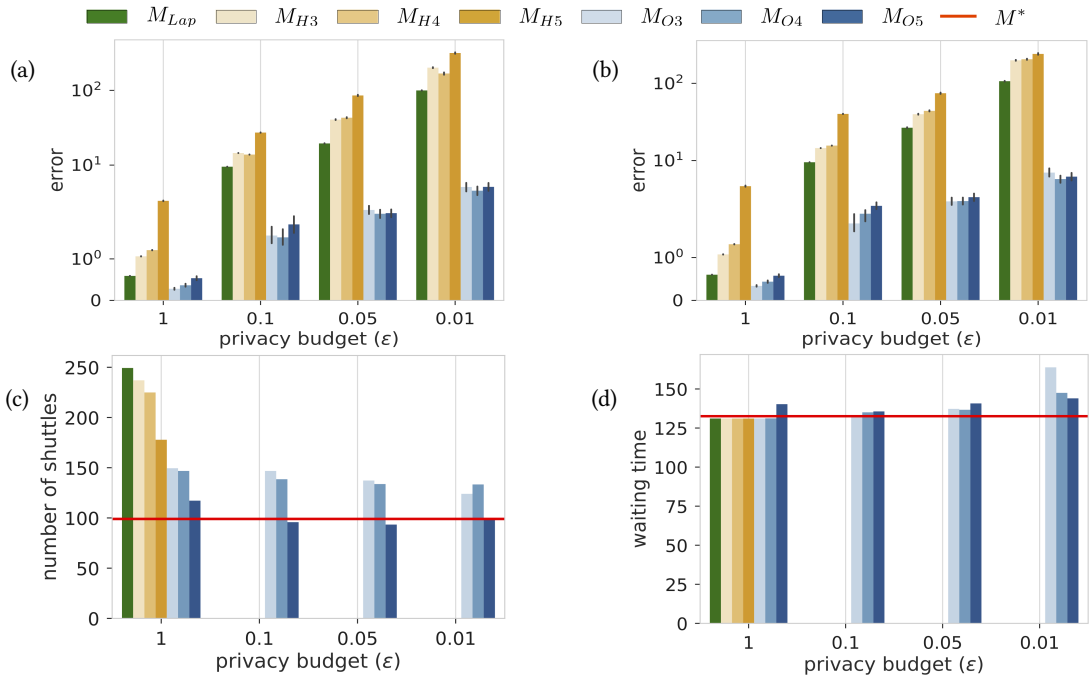

Figure 5: Average errors for the weekend (a) and the weekday (b) trip data sets, and MTSOP system cost (c) and average waiting time (d) reported at the varying of the privacy budget.

- $\mathcal{M}_{O3}$ uses three features: $\mathbf{F}_1$ and $\mathbf{F}_2$, and $\mathbf{F}_3$ The optimization step is therefore with objective Eq. (5, 6) and constraints (9–11) and (18);
- $\mathcal{M}_{O4}$ uses four features: $\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3$, and $\mathbf{F}_4$. The optimization step is therefore with objective Eq. (5–7) and constraints (9–14) and (18);
- $\mathcal{M}_{O5}$ uses all five features. The optimization step is therefore with objective Eq. (5–8) and constraints (9–18).

Hierarchical Mechanism. The hierarchical mechanism was also evaluated with three different sets of features. Mechanisms $\mathcal{M}_{H3}$, $\mathcal{M}_{H4}$, and $\mathcal{M}_{H5}$ generate trees of heights 3, 4, and 5 respectively. The root in $\mathcal{M}_{H3}$ represents all trip types, the next level represents the trip partitioning by time intervals, and each leaf is assigned to an individual trip types. $\mathcal{M}_{H4}$ extends $\mathcal{M}_{H3}$ with an second intermediate level for the zones. $\mathcal{M}_{H5}$ has a second intermediate level for zones and a third intermediate level for modes. $\mathcal{M}_{H3}$, $\mathcal{M}_{H4}$, and $\mathcal{M}_{H5}$ are thus the counterparts of $\mathcal{M}_{O3}$, $\mathcal{M}_{O4}$, and $\mathcal{M}_{O5}$ respectively. All the experiments with the hierarchical mechanism employ the heuristic used by Hay et al. [11]: If a node in T_1 has a non-positive value, then the node and all its successors are pruned. This heuristic was shown to reduce the error in sparse data sets [11]. Additionally, the privacy budget is distributed according to the geometric scheme from Cormode et al., which was shown to outperform the original uniform privacy budget allocation scheme.

Privacy Budgets and Execution Times. The mechanisms are evaluated for privacy budgets $\epsilon \in \{1.0, 0.1, 0.05, 0.01\}$, where smaller values for ϵ means increased privacy guarantees and hence more noise in the private counts. All experimental results are reported as the average of 50 runs. The CPU times to generate the private datasets are between 30.1 and 543 seconds for the $\mathcal{M}_H$ experiments, 9.9 and 52.8 seconds for $\mathcal{M}_{Lap}$, and 7.8 and 15.4 seconds for the $\mathcal{M}_O$ experiments, depending on the chosen privacy budget. Interestingly, the $\mathcal{M}_O$ experiments, despite the convex optimization step, can be faster than the $\mathcal{M}_{Lap}$ experiment. This is due to the huge volume of extra data points that $\mathcal{M}_{Lap}$ introduces.

Error Analysis. The first experiments measure the error introduced by a mechanism as the average distance between the exact and the private answers to the counting queries. The error of the mechanisms are depicted in Figure 5(a) for the *weekday* (high demand) and in Figure 5(b) for the *weekend* (low demand) data sets using various privacy budgets. The results are shown in log-scale.

For all privacy budgets, the hierarhical mechanism performs slightly worse than $\mathcal{M}_{Lap}$, with $\mathcal{M}_{H5}$ reporting the largest errors. Obviously, $\mathcal{M}_{H5}$ introduces more noise than $\mathcal{M}_{H4}$, $\mathcal{M}_{H3}$, and $\mathcal{M}_{Lap}$, as it splits the privacy budget into more levels. Overall, for the ODMTS case study, the hierarchical mechanism is not able to redistribute the additional noise effectively to obtain a higher accuracy.

In contrast, the CBDP mechanism always improve the Laplace mechanism and brings an order of magnitude improvement in accuracy over the Laplace and hierarchical mechanisms as soon as $\epsilon < 1.0$. This is especially striking with respect to the hierarchical mechanisms since both sets of mechanisms apply the same amount of noise.[2] This highlights the benefits of the optimization model in

[2]The magnitude of improvements remain the same when a uniform privacy budget allocation is used for the $\mathcal{M}_H$ mechanisms.

	Weekday			Weekend		
	shuttles	wtime (s)	itime (s)	shuttles	wtime (s)	itime (s)
$\mathcal{M}_{O5}$	**0.158**	0.040	**0.061**	**0.252**	**0.005**	**0.004**
$\mathcal{M}_{O4}$	0.483	**0.003**	0.196	0.843	0.024	0.049
$\mathcal{M}_{O3}$	0.510	0.028	0.145	0.891	0.024	0.051
$\mathcal{M}_{H5}$	0.788	0.028	0.483	2.647	0.025	0.150
$\mathcal{M}_{H4}$	1.262	0.007	0.578	3.694	0.026	0.159
$\mathcal{M}_{H3}$	1.394	0.029	0.586	3.704	0.024	0.172
$\mathcal{M}_{\text{Lap}}$	1.518	0.029	0.587	4.371	0.021	0.192
$\mathcal{O}(\pi, D_o)$	99	135.0	5130	48	127.8	4779

Table 1: Error on the system utilities obtained using private data sets generated with privacy budget $\epsilon = 1.0$.

the post-processing step. The discussion in the next section provides an in-depth study of the sources of the improvements observed in the $\mathcal{M}_O$ experiments.

A strength of the CBDP mechanisms is their ability to represent non-negativity constraints on the decision variables in the optimization model. In contrast, $\mathcal{M}_{H3}$, $\mathcal{M}_{H4}$, and $\mathcal{M}_{H5}$ produce 8285, 7006, and 11453 negative counts in average over all our experiments. The average values span from -2.3 (for the lowest privacy budget) to -171.6 (for the highest privacy budget), suggesting that the mechanism must generate leaf nodes with large positive counts to satisfy the constraints imposed by the counts in the hierarchy.

Analysis of the ODMTS Utility. Consider now the evaluation of mechanisms on the utility of the ODMTS, which is the most important metric in our context. The effectiveness of a mechanism is evaluated by considering a trip data set D and the private version $\tilde{D}$ it generates. The comparison first performs the fleet sizing over the original and private dataset, i.e., it computes $\pi = \mathcal{F}(D)$ and $\tilde{\pi} = \mathcal{F}(\tilde{D})$. Then, it evaluates the real-time operation configured with π and $\tilde{\pi}$ respectively on a real operational dataset D_o, i.e., it computes $\mathcal{O}(\tilde{\pi}, D_o)$ and $\mathcal{O}(\pi, D_o)$. These optimization models make it possible to evaluate the system cost under various mechanisms, as well as the performance of the ODMTS operations in terms of convenience.

Figure 5(c) present the results in terms of system costs. They use the number of shuttles as a proxy for costs. Since the cost of shuttle drivers represents about 50 percent of the total cost in the actual system and the experiments can only be performed for the 8am-10am time period because of the scalability of the Laplace and hierarchical mechanisms, this proxy can be used to compare the various mechanisms. Figure 5(d) present the results for the average waiting time per leg in seconds on the weekday dataset again. Both figures report results for various privacy budgets and the red lines report the value of $\mathcal{F}(D)$ and $\mathcal{O}(\pi, D)$ respectively. The mechanism results should thus be as close as possible to the red lines. Unfortunately, it was not possible to find a feasible fleet sizing (within a 24h limit) for the trip datasets produced by $\mathcal{M}_{Lap}$ and the $\mathcal{M}_H$ variants for privacy budgets smaller than 1 due to the loss of sparsity in these mechanisms. To provide some perspectives, the number of trips generated by $\mathcal{M}_{Lap}$ for these privacy budgets are 320, 940.40, 656,577.40, and 3,312,350.00. For $\mathcal{M}_{H_5}$, they are 839,150.50, 1,698,770.00, and 8,543,366.50.

Figure 5(c) shows that the CBDP mechanism better preserves the system costs compared to other mechanisms. In general, all mechanisms produce over-sized fleets but the cost increases of the Laplace and hierarchical mechanisms are very significant. In contrast, when the privacy budget decreases, mechanism $\mathcal{M}_{O5}$ exactly matches, or slightly underestimates, the number of shuttles. These results show that $\mathcal{M}_{O5}$ produces a small increase in cost.

Figure 5(d) shows that all the mechanisms preserve the waiting times with great accuracy. This is not surprising since the mechanisms almost always have over-sized fleets. When $\mathcal{M}_{O5}$ slightly underestimates the fleet size for the smallest privacy budget, a slight increase in waiting time is observed.

Table 1 tabulates the ODMTS utilities in these experiments for the weekday and weekend datasets and privacy budget $\epsilon = 1.0$. The last row reports the fleet sizing (shuttles), the average waiting time per trip in seconds (wtime), and the average shuttles idle time in seconds (itime), defined as the total amount of time during which a vehicle is serving no passenger. The other rows show the relative error between the true utility f^* of the ODMTS and its counterpart $\tilde{f}$ for a given mechanism. The highlighted values denote the most accurate utilities across the mechanisms. The table clearly shows that $\mathcal{M}_{O_5}$ produces results with smaller relative errors for all metrics. For the weekday setting, $\mathcal{M}_{O_5}$ increases the fleets size by 16% and the waiting time by 4% for this privacy budget. In contrast, the hierarchical mechanisms increase the number of vehicles in the system by 79% to 139% and the Laplace mechanism by 152%. *The benefits of the CBDP mechanism are thus substantial.*

11 DISCUSSION

This section provides an in-depth analytical comparison between the hierarchical mechanisms and the CBDP mechanisms. It is important to mention that the hierarchical mechanisms were designed for a different purpose: They target a situation where users submit a sequence of queries $\langle Q_1, \ldots, Q_n \rangle$ and the hierarchical mechanism is used to ensure consistency across the queries, while preserving the overall accuracy. The CBDP mechanism in contrast was designed for producing private data sets in mobility: The application is only interested in the counts of the possible trip types but the mechanism adds additional queries to improve the accuracies of these counts given the large universe and the sparsity of the datasets. However, as shown earlier, given a feature set $\mathscr{F} = \{\mathbf{F}_1, \ldots, \mathbf{F}_k\}$, it is possible to solve the ODMTS problem with $\mathcal{M}_H$ by constructing

a hierarchy where $\mathbf{F}_2$ describes the root node, $\mathbf{F}_1$ the leaf nodes, and the node for intermediate level i are the elements of the feature set $\mathbf{F}_{k-i+1}$. Thus, both methods make use of an optimization step to post-process the counting queries, although $\mathcal{M}_H$ "solves" a relaxed version of the problem using a closed-form solution (relaxing the non-negativity constraints), while $\mathcal{M}_O$ solves an actual convex optimization problem. There are three key aspects that differentiate the CBDP from the hierarchical method:

1. The Problem Structure. The CBDP and hierarchical mechanisms imposed a fundamentally different structure on the constraints of the optimization model: flat (F) or hierarchical (H). The flat structure imposed by $\mathcal{M}_O$ allows each element of a feature set to be related via a constraint to any set of elements of any other feature set. The hierarchical structure used by $\mathcal{M}_H$ imposes a hierarchy on the elements of the feature set so that children counts are related to the parent counts. This has an important consequence on the number of total queries the mechanisms need to answer. In the flat representation, the mechanism requires at most $\sum_{\mathbf{F}_i \in \mathscr{F}} |\mathbf{F}_i|$ queries while, in the hierarchical representation, a mechanism requires $\prod_{\mathbf{F}_i \in \mathscr{F}} |\mathbf{F}_i|$ queries. Imposing a hierarchical structure has also an important consequence on the amount of decision variables and constraints of the optimization model. The program requires a variable for each count associated with a node, i.e., $\prod_{\mathbf{F}_i \in \mathscr{F}} |\mathbf{F}_i|$ variables. In contrast, a flat structure requires $\sum_{\mathbf{F}_i \in \mathscr{F}} |\mathbf{F}_i|$ variables. The same consideration holds for constraints. These quantities negatively and dramatically affect the performance of a hierarchical organization for the optimization method, hindering its scalability.

2. Non-Negative Variables. The CBDP mechanism naturally handles the negative outputs from the Laplace mechanism via *post-processing*. In contrast, the hierarchical mechanism's closed-form solution ignores the nonnegativity constraints. Such property is especially important for counting queries on sparse datasets, as in the ODMTS case study.

3. Weighted Distance. The CBDP mechanism also optimizes a weighted L_2-norm, which ensures that each feature contributes equally to the objective. This is important since they all are partitions of the same universe.

Impact Evaluation. These three aspects are now evaluated on the case study. In addition to $\mathcal{M}_{O5}$ and $\mathcal{M}_{H5}$, the following variants of $\mathcal{M}_{O5}$ are considered:

- $\mathcal{M}_{O5}(a)$ uses a hierarchical structure in the constraints, non-negative decision variables, and uniform weights for the terms in the objective.
- $\mathcal{M}_{O5}(b)$ uses a hierarchical structure in the constraints, non-negative decision variables, and the weighting scheme of $\mathcal{M}_{O5}$.
- $\mathcal{M}_{O5}(c)$ uses a flat structure, non-negative decision variables, and a uniform weighting scheme.

Table 2 tabulates the errors introduced by the mechanisms as in Figures 5(a–b), i.e., the average distance between the exact and the private answers to the counting queries. The columns *S*, *P*, and *W*, respectively, denote the type of structure adopted by the mechanism (F or H), whether the optimization uses non-negative valued variables, and whether a weighting scheme is used for the optimization. The results clearly indicate that all three aspects are critical for the required accuracy and demonstrate the superiority of the combination adopted by $\mathcal{M}_{O5}$. The second line shows that the performance of the hierarchical mechanism significantly degrades when the privacy budget decreases: The magnitude of its errors are more than 5, 14, and 118 times larger from weekdays than $\mathcal{M}_{O5}$ for privacy budgets 1.0, 0.1, and 0.01. Mechanism $\mathcal{M}_{O5}(a)$ shows that adding the non-negativity constraints is helpful in slowing down the performance degradation as the privacy budget decreases but the errors remain more than an order of magnitude larger than $\mathcal{M}_{O5}$. Mechanism $\mathcal{M}_{O5}(b)$ indicates the importance of a flat structure in the optimization: For privacy budgets 1.0 and 0.1, the errors introduced by a hierarchical structure are an order of magnitude larger than those with a flat structure. Finally, mechanism $\mathcal{M}_{O5}(c)$ demonstrates the benefits of the weighted objective as the privacy budget decreases: For a budget of 0.01, the weighting scheme decreases the errors by a factor of about 2.5 in weekdays. This analysis clearly demonstrates the benefits of the CBDP mechanism for mobility applications.

	Prop	Privacy budget					
		Weekday			Weekend		
	S P W	1.0	0.1	0.01	1.0	0.1	0.01
$\mathcal{M}_{O5}$	F ✓ ✓	**0.91**	**3.87**	**6.01**	**0.86**	**2.69**	**5.77**
$\mathcal{M}_{H5}$	H ✗ ✗	5.11	56.6	715	4.57	50.9	728
$\mathcal{M}_{O5}(a)$	H ✓ ✗	35.6	44.2	108	42.2	59.9	120.5
$\mathcal{M}_{O5}(b)$	H ✓ ✓	13.4	15.7	18.9	13.4	14.5	19.6
$\mathcal{M}_{O5}(c)$	F ✓ ✗	1.50	5.55	15.2	1.34	4.27	18.4

Table 2: Analysis of the Key Features of CBDP.

12 CONCLUSIONS

This paper introduced Constraint-Based Differential Privacy (CBDP), an approach to differential privacy which aims at releasing private mobility data set for complex optimization tasks. CBDP casts the private data set release as an optimization problem that redistributes the noise introduced by a randomized mechanism to satisfy fundamental constraints of the original data. The optimization model minimizes a weighted L_2-norm between the noisy counts generated by the Laplace mechanism and post-processed counts that satisfy constraints imposed by the features of the application.

CBDP has been evaluated on several mobility datasets for the city of Ann Arbor, MI, and the design and operations of an On-Demand Multimodal Transit System (ODMTS). Experimental results show that CBDP improves the accuracy of existing approaches (the Laplace mechanism and hierarchical mechanisms) by an order of magnitude and results in negligible utility losses compared to an ODMTS designed with the actual data. These results are promising and indicates that CBDP has the potential to become an important tool at the intersection of privacy, mobility, and optimization.

Although the paper focused on the applicability of CBDP to ODMTS, the proposed mechanism is general and can be used for other applications where data features and/or knowledge of the problem of interest is available.

ACKNOWLEDGMENTS

This work was partially supported by the Michigan Institute of Data Science and the Seth Bonder Foundation.

REFERENCES

[1] Alastair R Beresford and Frank Stajano. 2003. Location privacy in pervasive computing. *IEEE Pervasive computing* 2, 1 (2003), 46–55.

[2] Graham Cormode, Cecilia Procopiuc, Divesh Srivastava, Entong Shen, and Ting Yu. 2012. Differentially private spatial decompositions. In *Data engineering (ICDE), 2012 IEEE 28th international conference on*. IEEE, 20–31.

[3] Graham Cormode, Magda Procopiuc, Divesh Srivastava, and Thanh TL Tran. 2011. Differentially private publication of sparse data. *arXiv preprint arXiv:1103.0825* (2011).

[4] Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. 2013. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports* 3 (2013), 1376.

[5] Bolin Ding, Marianne Winslett, Jiawei Han, and Zhenhui Li. 2011. Differentially private data cubes: optimizing noise sources and consistency. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*. ACM, 217–228.

[6] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *TCC*, Vol. 3876. Springer, 265–284.

[7] Cynthia Dwork, Aleksandar Nikolov, and Kunal Talwar. 2015. Efficient algorithms for privately releasing marginals via convex relaxations. *Discrete & Computational Geometry* 53, 3 (2015), 650–673.

[8] Cynthia Dwork and Aaron Roth. 2013. The algorithmic foundations of differential privacy. *Theoretical Computer Science* 9, 3-4 (2013), 211–407.

[9] Ferdinando Fioretto and Pascal Van Hentenryck. 2018. Constrained-based Differential Privacy: Releasing Optimal Power Flow Benchmarks Privately. In *Proceedings of the International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR)*.

[10] Marco Gruteser and Xuan Liu. 2004. Protecting privacy, in continuous location-tracking applications. *IEEE Security & Privacy* 2, 2 (2004), 28–34.

[11] Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. 2010. Boosting the accuracy of differentially private histograms through consistency. *Proceedings of the VLDB Endowment* 3, 1-2 (2010), 1021–1032.

[12] Justin Hsu, Marco Gaboardi, Andreas Haeberlen, Sanjeev Khanna, Arjun Narayan, Benjamin C Pierce, and Aaron Roth. 2014. Differential privacy: An economic method for choosing epsilon. In *Computer Security Foundations Symposium (CSF), 2014 IEEE 27th*. IEEE, 398–410.

[13] Dong Huang, Shuguo Han, Xiaoli Li, and Philip S Yu. 2015. Orthogonal mechanism for answering batch queries with differential privacy. In *Proceedings of the 27th International Conference on Scientific and Statistical Database Management*. ACM, 24.

[14] Fragkiskos Koufogiannis, Shuo Han, and George J Pappas. 2015. Optimality of the laplace mechanism in differential privacy. *arXiv preprint arXiv:1504.00065* (2015).

[15] John Krumm. 2009. A survey of computational location privacy. *Personal and Ubiquitous Computing* 13, 6 (2009), 391–399.

[16] Chao Li, Michael Hay, Gerome Miklau, and Yue Wang. 2014. A data-and workload-aware algorithm for range queries under differential privacy. *Proceedings of the VLDB Endowment* 7, 5 (2014), 341–352.

[17] Tiancheng Li and Ninghui Li. 2009. On the tradeoff between privacy and utility in data publishing. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 517–526.

[18] Yang D Li, Zhenjie Zhang, Marianne Winslett, and Yin Yang. 2011. Compressive mechanism: Utilizing sparse representation in differential privacy. In *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*. ACM, 177–182.

[19] Arthur Maheo, Phlip Kilby, and Pascal Van Hentenryck. 2017. Benders Decomposition for the Design of a Hub and Shuttle Public Transit System. *Transportation Science* (2017). To appear.

[20] Arkadi Nemirovski. 2004. *Interior Point Polynomial Time Methods IN Convex Programming*. Technical Report ISYE 8813. Georgia Institute of Technology, Atlanta, GA.

[21] Wahbeh Qardaji, Weining Yang, and Ninghui Li. 2013. Understanding hierarchical methods for differentially private histograms. *Proceedings of the VLDB Endowment* 6, 14 (2013), 1954–1965. http://www.vldb.org/pvldb/vol6/p1954-qardaji.pdf

[22] Wahbeh Qardaji, Weining Yang, and Ninghui Li. 2014. PriView: practical differentially private release of marginal contingency tables. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM, 1435–1446.

[23] Salil Vadhan. 2017. The complexity of differential privacy. In *Tutorials on the Foundations of Cryptography*. Springer, 347–450.

[24] Xiaokui Xiao, Guozhang Wang, and Johannes Gehrke. 2011. Differential privacy via wavelet transforms. *IEEE Transactions on Knowledge and Data Engineering* 23, 8 (2011), 1200–1214.