Enhancing Scalability of Metric Differential Privacy via Secret Dataset Partitioning and Benders Decomposition

Chenxi Qiu
Department of Computer Science and Engineering, University of North Texas
chenxi.qiu@unt.edu

Abstract

Metric Differential Privacy (mDP) extends the concept of Differential Privacy (DP) to serve as a new paradigm of data perturbation. It is designed to protect secret data represented in general metric space, such as text data encoded as word embeddings or geo-location data on the road network or grid maps. To derive an optimal data perturbation mechanism under mDP, a widely used method is *linear programming (LP)*, which, however, might suffer from a polynomial explosion of decision variables, rendering it impractical in large-scale mDP.

In this paper, our objective is to develop a new computation framework to enhance the scalability of the LP-based mDP. Considering the connections established by the mDP constraints among the secret records, we partition the original secret dataset into various subsets. Building upon the partition, we reformulate the LP problem for mDP and solve it via *Benders Decomposition*, which is composed of two stages: (1) a master program to manage the perturbation calculation across subsets and (2) a set of subproblems, each managing the perturbation derivation within a subset. Our experimental results on multiple datasets, including geo-location data in the road network/grid maps, text data, and synthetic data, underscore our proposed mechanism's superior scalability and efficiency.

1 Introduction

Among the array of data privacy protection mechanisms, *data perturbation* has emerged as a widely employed technique to protect individuals' information. Data perturbation involves intentionally introducing noise into a database, rendering individual information unreadable by unauthorized users even in the event of data breaches on the server side. Particularly, *Differential Privacy (DP)* [Dwork *et al.*, 2006] has become a paradigm of choice for data perturbation due to its strong and theoretically provable privacy guarantees.

In its original definition, DP requires data perturbation to maintain a uniform *indistinguishability* level for the queries of the *neighboring databases* [Dwork *et al.*, 2006]. Specifically, the classification of "neighboring databases" relies on their *Hamming distance*, i.e., two databases are considered "neighbors" if they differ by at most one record. However, this definition limits DP's applicability in the domains where data similarity is evaluated by other distance metrics, and also varying degrees of indistinguishability are necessitated by nonbinary distance values among neighbors. Recognizing these constraints, DP has been extended to *metric DP (mDP)* [Imola *et al.*, 2022], which generalizes the classification of "neighboring databases" by accounting for the diverse underlying distance metric spaces.

Related work of mDP. mDP originated in the domain of geo-location privacy protection [Andrés *et al.*, 2013], requiring "*geo-indistinguishability*" for each pair of locations with the Euclidean distance lower than a predetermined threshold. In other words, mDP defines "neighboring locations" based on Euclidean distance, diverging from the original DP which relies on Hamming distance. Over time, mDP has been explored across a spectrum of metric choices, including Manhattan distance [Chatzikokolakis *et al.*, 2013], Hyperbolic distance [Feyisetan *et al.*, 2019], Haversine distance [Pappachan *et al.*, 2023], Word Mover's distance [Fernandes *et al.*, 2019], and others [Feyisetan and Kasiviswanathan, 2021].

Compared to the conventional DP, the optimization of mDP poses additional challenges due to the diverse sensitivity of utility loss to data perturbation in general distance metric spaces [Qiu *et al.*, 2022a]. A commonly employed approach is to discretize both the secret dataset and the corresponding perturbed data domain into finite sets [Imola *et al.*, 2022], which allows for explicit measurement of the utility loss caused by each perturbation choice. In this case, the problem of optimizing the perturbation distribution of each secret record can be formulated as a *linear programming (LP)* problem [Fawaz and Shin, 2014], of which the objective is to minimize the expected utility loss caused by data perturbation while satisfying the mDP constraints for each pair of neighboring records.

However, the LP formulation of mDP typically requires $O(N^2)$ decision variables and $O(N^2K)$ linear constraints [Bordenabe *et al.*, 2014], where N and K are the size of the secret dataset and perturbed dataset, respectively. The high complexity of LP makes it hard to apply in a large-scale mDP. As illustrated by Fig. 1, most current LP-based mDP works have to limit their applications to small-scale secret datasets

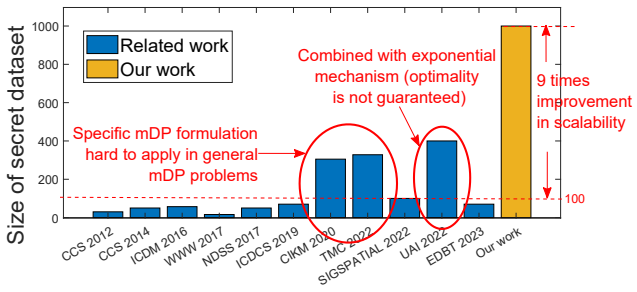


Figure 1: Comparison of the secret dataset size in the related LP-based mDP works and our work.
CCS 2012 [Shokri *et al.*, 2012], CCS 2014 [Fawaz and Shin, 2014], ICDM 2016 [Wang *et al.*, 2016], WWW 2017 [Wang *et al.*, 2017], NDSS 2017 [Yu *et al.*, 2017], ICDCS 2019 [Qiu and Squicciarini, 2019], CIKM [Qiu *et al.*, 2020], TMC 2022 [Qiu *et al.*, 2022a], SIGSPATIAL 2022 [Qiu *et al.*, 2022b], UAI 2022 [Imola *et al.*, 2022], EDBT 2023 [Pappachan *et al.*, 2023].

(i.e., up to 100 records in the secret dataset [Chatzikokolakis *et al.*, 2015]). While some recent efforts [Qiu *et al.*, 2020; Qiu *et al.*, 2022a] have managed to extend the size of secret datasets to around 300 records using Dantzig-Wolfe decomposition and column generation algorithms, their focus has been on specific LP formulations where the mDP constraints are imposed for all pairs of secret records, simplifying the initialization of column generation. This specific LP formulation, however, is hard to apply in general mDP problems where the constraints are required only for neighboring records. Alternative approaches like [Imola *et al.*, 2022] integrate LP with the exponential mechanism to enhance scalability, which, however, comes at the cost of sacrificing the optimality of mDP.

Our main contributions. Our primary aim is to enhance the scalability of the LP-based mDP by introducing a new computation framework. We first construct an *mDP graph* to describe the mutual mDP constraints among secret records and then partition the entire secret dataset into well-balanced subsets based on the mDP graph (**Contribution 1**). Building upon this partition, we then reformulate the LP and solve it using *Benders Decomposition* (**Contribution 2**), which consists of two stages: (1) a master program to manage the perturbation distribution calculations across subsets and (2) a set of subproblems, each deriving the perturbation distributions of records within a subset. The problems in both stages exhibit a relatively small scale, enabling efficient solutions. This efficiency facilitates the iterative derivation of a near-optimal solution for the original LP through the communication between the two stages.

Finally, we test the performance of the new computation framework using geo-location data (in the road network and grid maps), text embeddings, and synthetic datasets, with a comparison of several benchmarks (**Contribution 3**). The experimental results demonstrate that our new computation framework can derive the optimal mDP for the secret dataset with 1,000 records, marking approximately 9 times improvement in scalability compared to the state-of-the-art methods for general mDP problems, as shown in Fig. 1.

2 Preliminaries

In this part, we introduce the preliminary knowledge of mDP in **Section 2.1** and its LP computation framework in **Section 2.2**. Table 4 in [Qiu, 2024] lists the main notations used throughout this paper.

2.1 Metric DP

In general, a data perturbation mechanism can be represented as a *probabilistic function* Q: $\mathcal{R} \to \mathcal{O}$, of which the domain $\mathcal{R}$ is users' *secret dataset* and the range $\mathcal{O}$ is the *perturbed dataset*. Given the underlying data features of $\mathcal{R}$, a metric $d : \mathcal{R}^2 \mapsto \mathbb{R}$ is defined to measure the *distance* between any two records $r_i, r_j \in \mathcal{R}$, where the distance between r_i and r_j is denoted by $d_{i,j}$.

Definition 2.1. *(Neighboring records) Given the secret dataset $\mathcal{R}$ and its distance measure d, any pair of records $r_i, r_j \in \mathcal{R}$ are called* neighboring records *if their distance $d_{i,j} \leq \eta$, where $\eta > 0$ is a pre-determined threshold.*

In what follows, we use $\mathcal{E} = \{(r_i, r_j) \in \mathcal{R}^2 \,|d_{i,j} \leq \eta\}$ to represent the whole set of neighboring records in $\mathcal{R}$.

Definition 2.2. *(Metric DP) For each pair of neighboring records, $(r_i, r_j) \in \mathcal{E}$, ϵ-mDP ensures that the probability distributions of their perturbed data $Q(r_i)$ and $Q(r_j)$ are sufficiently close so that it is hard for an attacker to distinguish r_i and r_j according to the probability distributions of $Q(r_i)$ and $Q(r_j)$*

$$\frac{\Pr\{Q(r_i) \in O\}}{\Pr\{Q(r_j) \in O\}} \leq e^{\epsilon d_{i,j}}, \ \forall O \subseteq \mathcal{O}. \tag{1}$$

Here, $\epsilon > 0$ is called the privacy budget*, reflecting how much information can be disclosed from the perturbed data, i.e., lower ϵ implies a higher privacy level.*

2.2 Linear Programing-based Approaches

Considering the computational challenges of optimizing a perturbation function Q defined on a continuous domain $\mathcal{R}$ and a continuous range $\mathcal{O}$, it is common practice to discretize both $\mathcal{R}$ and $\mathcal{O}$ to finite sets [Imola *et al.*, 2022]. In this case, the function Q can be represented as a stochastic *perturbation matrix* $\mathbf{Z} = \{z_{i,k}\}_{(r_i,o_k)\in\mathcal{R}\times\mathcal{O}}$, where each entry $z_{i,k}$ represents the probability of selecting $o_k \in \mathcal{O}$ as the perturbed record given the real record $r_i \in \mathcal{R}$. As such, the mDP constraints formulated in Eq. (1) can be rewritten as the following linear constraints

$$z_{i,k} - e^{\epsilon d_{i,j}} z_{j,k} \leq 0, \ \forall o_k \in \mathcal{O}, \forall (r_i, r_j) \in \mathcal{E}. \tag{2}$$

Additionally, the sum probability of perturbed record $o_k \in \mathcal{O}$ for each real record r_i should be equal to 1 (the *unit measure* of probability theory [Stroock, 2010]), i.e.,

$$\sum_{o_k \in \mathcal{O}} z_{i,k} = 1, \ \forall r_i \in \mathcal{R}. \tag{3}$$

We let $\mathbf{z}_i = [z_{i,1}, ..., z_{i,K}]$ $(i = 1, ..., K)$ denote the record r_i's *perturbation vector*, i.e., the probability distribution of its perturbed records. We let $c_{i,k}$ denote the utility loss caused by

the perturbed record o_k when the real record is r_i. Then, the expected utility loss caused by the perturbation matrix $\mathbf{Z}$ can be represented by $\sum_{i=1}^{N} \mathbf{c}_i \mathbf{z}_i^\top$, where $\mathbf{c}_i = [p_i c_{i,1}, ..., p_i c_{i,K}]$ and p_i denotes the prior probability of the record being r_i.

Consequently, the objective of the *perturbation matrix optimization (PMO)* problem is to minimize the expected data utility loss $\sum_{i=1}^{N} \mathbf{c}_i \mathbf{z}_i^\top$ while satisfying the constraints of both mDP and the probability unit measure, which can be formulated as the following LP problem:

$$\min \quad \sum_{i=1}^{N} \mathbf{c}_i \mathbf{z}_i^\top \tag{4}$$

$$\text{s.t.} \quad \text{mDP constraints (Eq. (2)) are satisfied } \forall r_i \in \mathcal{R} \tag{5}$$

$$\text{Unit measure (Eq. (3)) is satisfied } \forall r_i \in \mathcal{R} \tag{6}$$

$$0 \leq z_{i,k} \leq 1, \forall (r_i, o_k) \in \mathcal{R} \times \mathcal{O}. \tag{7}$$

The decision variables in the LP problem in Eq. (4)-(7) are the perturbation matrix $\mathbf{Z}$, including $O(NK)$ decision variables (entries), constrained by $O(N^2K)$ linear constraints. Such a high complexity makes this LP computation framework hard to apply in large-scale mDP applications [Pappachan *et al.*, 2023].

3 Methodology

In this section, we present our approach to enhance the scalability of the mDP calculation via LP decomposition.

We first present the computational framework in **Section 3.1**, followed by the detailed descriptions of its two components: *Secret dataset partitioning* and *Benders decomposition (BD)*. In our framework, the secret dataset partitioning precedes BD, but since the objective for partitioning the secret data is contingent on the complexity analysis of BD, we introduce BD in **Section 3.2** and its time complexity analysis in **Section 3.3** before delving into secret dataset partitioning in **Section 3.4**.

3.1 Computation Framework

In general, the decomposition of an optimization problem highly depends on how its decision variables are coupled by constraints [Palomar and Chiang, 2006]. A common strategy involves partitioning the original large-scale problem into a set of subproblems with smaller sizes. Within each subproblem, the decision variables are strongly linked by constraints, while the connections between decision variables across different subproblems are comparatively weaker.

In PMO, the decision variables are the *perturbation vectors* of secret records, $\mathbf{z}_i = [z_{i,1}, ..., z_{i,K}]$ $(i = 1, ..., K)$, linked by the mDP constraints (Eq. (2)). Here, we describe how the perturbation vectors of each pair of records are coupled by the mDP constraints in $\mathcal{R}$ by an undirected graph, called the *mDP graph* (**Definition 3.1**):

Definition 3.1. *(mDP graph) The mDP graph $\mathcal{G}$ is defined as the ordered pair $\mathcal{G} = (\mathcal{R}, \mathcal{E})$, where the node set $\mathcal{R}$ is the secret dataset and the edge set $\mathcal{E}$ is the whole set of neighboring records in $\mathcal{R}$. The weight assigned to each edge (r_i, r_j) is the distance $d_{i,j}$ between the two records.*

As Fig. 2 shows, given the mDP graph $\mathcal{G}$, our intuitive idea is to first partition $\mathcal{G}$ into M subgraphs: $\mathcal{G}_l = (\mathcal{R}_1, \mathcal{E}_1), ..., \mathcal{G}_M = (\mathcal{R}_M, \mathcal{E}_M)$, such that the nodes within each subgraph are *strongly connected* by the mDP constraints (i.e., the perturbation vector of one node exerts a significant mDP constraint on the other), and the nodes across different subgraphs are *weakly connected* by the mDP constraints. Subsequently, a subproblem, denoted by Sub_l, can be formulated to determine the perturbation vectors of the secret records within each subgraph $\mathcal{G}_l$ $(l = 1, ..., N)$. Here, we use $\mathbf{z}_{\mathcal{R}_l} = \{\mathbf{z}_i, | r_i \in \mathcal{R}_l\}$ to denote the perturbation vectors of the set of records in $\mathcal{R}_l$.

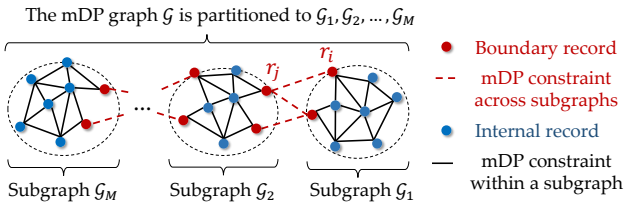


Figure 2: Computational framework.

Next, we introduce Property 3.1 and Property 3.2 of the mDP graph to facilitate the computation of the PMO problem. The detailed proofs of the two properties can be found in Section B.1 and Section B.2 in [Qiu, 2024].

Property 3.1. *(Independent computation of mDP components) Suppose that the mDP graph $\mathcal{G}$ has m components, denoted by $\mathcal{C}_l = (\mathcal{R}_l, \mathcal{E}_l)$ $(l = 1, ..., m)$. Here, each component is a connected subgraph that is not part of any larger connected subgraph [Wilson, 1986]. Consider a subproblem Sub_l formulated to determine the optimal perturbation vectors for each $\mathcal{R}_l$, where only the mDP constraints in $\mathcal{E}_l$ are considered. In this case, we can solve $\mathrm{Sub}_1, ..., \mathrm{Sub}_M$ independently, and their collective optimal solutions form the optimal solution for PMO.*

Property 3.2. *Given that two records r_i and r_j (not necessarily neighbors) are connected by a path in the mDP graph $\mathcal{G}$, of which the* shortest path distance *(i.e., the sum weight of the edges in the path) is $D_{i,j}$, then $z_{i,k}$ and $z_{j,k}$ are restricted by the following constraints:*

$$z_{i,k} - e^{\epsilon D_{i,j}} z_{j,k} \leq 0, \ \forall o_k \in \mathcal{O}. \tag{8}$$

In what follows, we restrict our attention to the case where the mDP graph $\mathcal{G}$ has a single component, considering the case of multiple components as straightforward to generalize according to Property 3.1. When $\mathcal{G}$ is a single-component graph, although dividing $\mathcal{G}$ facilitates the parallel computation of the major perturbation vectors within each subset, it remains imperative to jointly derive the perturbation vectors that are linked by mDP constraints across multiple subsets. For instance, in Fig. 2, the perturbation vector $\mathbf{z}_i$ of r_i in $\mathcal{G}_1$ must adhere to mDP constraints with the perturbation vector $\mathbf{z}_j$ of r_j in $\mathcal{G}_2$. Based on whether the records in each $\mathcal{R}_l$ are adjacent to records in other subsets by mDP constraints, we classify the records in each $\mathcal{R}_l$ into either "*boundary records*" or "*internal records*", as defined in **Definition 3.2**:

Definition 3.2. *Each $r_i \in \mathcal{R}_l$ is a **boundary record** if it has at least a neighbor r_j in another subset $\mathcal{R}_n$ $(n \neq l)$ in the mDP graph $\mathcal{G}$; otherwise, r_i is an **internal record**.*

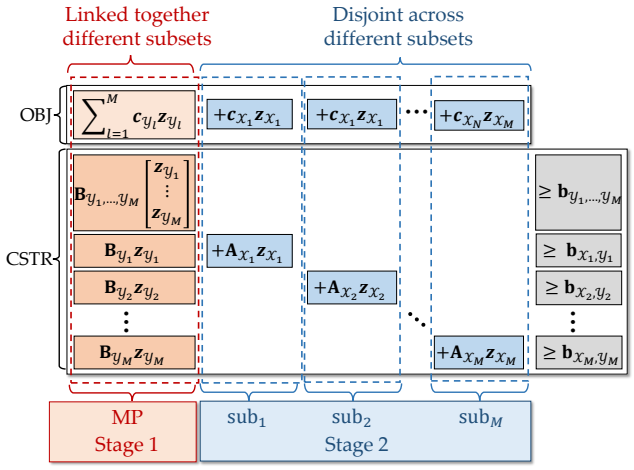


Figure 3: Block ladder structure of the PMO formulation.

We use $\mathcal{X}_l$ and $\mathcal{Y}_l$ ($\mathcal{X}_l, \mathcal{Y}_l \subseteq \mathcal{R}_l$) to represent the *internal record set* and the *boundary record set* in $\mathcal{R}_l$, respectively. We let $\mathbf{z}_{\mathcal{X}_l}$ and $\mathbf{z}_{\mathcal{Y}_l}$ denote the perturbation vectors of $\mathcal{X}_l$ and $\mathcal{Y}_l$, i.e., $\mathbf{z}_{\mathcal{X}_l} = \{\mathbf{z}_i \,|r_i \in \mathcal{X}_l\}$ and $\mathbf{z}_{\mathcal{Y}_l} = \{\mathbf{z}_i \,|r_i \in \mathcal{Y}_l\}$.

The PMO problem reformulation. After categorizing the records in $\mathcal{R}_l$ into $\mathcal{X}_l$ and $\mathcal{Y}_l$ ($l = 1, ..., M$), the original PMO formulated in Eq. (4)–(7) can be rewritten in a *block ladder structure*, as shown in Fig. 3:

(1) The **objective (OBJ)** $\sum_{i=1}^{N} \mathbf{c}_i \mathbf{z}_i$ is rewritten as the sum of

- $\sum_{l=1}^{M} \mathbf{c}_{\mathcal{Y}_l} \mathbf{z}_{\mathcal{Y}_l}$ in the red block, representing the data utility loss of the boundary records in $\mathcal{Y}_1, ..., \mathcal{Y}_M$, and
- $\mathbf{c}_{\mathcal{X}_l} \mathbf{z}_{\mathcal{X}_l}$ ($l = 1, ..., M$) in the blue blocks, representing the data utility loss of the internal records in $\mathcal{X}_l$;

(2) The **constraints (CSTR)** includes

- $\mathbf{B}_{\mathcal{Y}_1,...,\mathcal{Y}_M} \begin{bmatrix} \mathbf{z}_{\mathcal{Y}_1} \\ \vdots \\ \mathbf{z}_{\mathcal{Y}_M} \end{bmatrix} \geq \mathbf{b}_{\mathcal{Y}_1,...,\mathcal{Y}_M}$, including the mDP constraints that connect the boundary perturbation vectors $\mathbf{z}_{\mathcal{Y}_1}, ..., \mathbf{z}_{\mathcal{Y}_M}$ across $\mathcal{R}_1$, ..., $\mathcal{R}_M$, and the unit measure constraints of each $\mathbf{z}_{\mathcal{Y}_l}$ ($l = 1, ..., M$),
- $\mathbf{B}_{\mathcal{Y}_l} \mathbf{z}_{\mathcal{Y}_l} + \mathbf{A}_{\mathcal{X}_l} \mathbf{z}_{\mathcal{X}_l} \geq \mathbf{b}_{\mathcal{X}_l,\mathcal{Y}_l}$ ($l = 1, ..., M$), including the mDP constraints between the perturbation vectors within $\mathcal{R}_l$ (including both $\mathcal{X}_l$ and $\mathcal{Y}_l$), and their unit measure constraints.

Such block ladder structure lends the reformulated PMO well to *Benders decomposition (BD)* [Rahmaniani *et al.*, 2017]. Due to the limit of space, we list the detailed formulations of the coefficient matrices $\mathbf{B}_{\mathcal{Y}_1,...,\mathcal{Y}_M}$, $\mathbf{B}_{\mathcal{Y}_l}$, $\mathbf{A}_{\mathcal{X}_l}$, and coefficient vectors $\mathbf{b}_{\mathcal{Y}_1,...,\mathcal{Y}_M}$, and $\mathbf{b}_{\mathcal{X}_l,\mathcal{Y}_l}$ in **Section A.2** in [Qiu, 2024].

3.2 Benders Decomposition

BD is composed of two stages, a **master program (MP)** to derive the perturbation vectors of all the **boundary records**, and a set of **subproblems**, labeled as Sub_l ($l = 1, ..., M$), to derive the perturbation vectors of the **internal records** in each subset $\mathcal{R}_l$.

Stage 1: Master program. The MP derives the boundary records' perturbation vectors $\mathbf{z}_{\mathcal{Y}_1}, ..., \mathbf{z}_{\mathcal{Y}_M}$ and replaces the data utility loss of the internal records in each $\mathcal{X}_l$ by a single decision variable w_l, i.e., $w_l = \mathbf{c}_{\mathcal{X}_l} \mathbf{z}_{\mathcal{X}_l}$. The MP is formally formulated as the following LP problem

$$\min \quad \sum_{l=1}^{M} \mathbf{c}_{\mathcal{Y}_l} \mathbf{z}_{\mathcal{Y}_l} + \sum_{l=1}^{M} w_l \tag{9}$$

$$\text{s.t.} \quad \mathbf{B}_{\mathcal{Y}_1,...,\mathcal{Y}_M} \begin{bmatrix} \mathbf{z}_{\mathcal{Y}_1} \\ \vdots \\ \mathbf{z}_{\mathcal{Y}_M} \end{bmatrix} \geq \mathbf{b}_{\mathcal{Y}_1,...,\mathcal{Y}_M} \tag{10}$$

$$\mathcal{H} : \text{Cut set of } \mathbf{z}_{\mathcal{Y}_1}, ..., \mathbf{z}_{\mathcal{Y}_M}, w_1, ..., w_M \tag{11}$$

$$\mathbf{z}_{\mathcal{Y}_l} \geq \mathbf{0}, \; l = 1, ..., M. \tag{12}$$

where each *cut* in $\mathcal{H}$ is a *linear inequality* of the decision variables $\mathbf{z}_{\mathcal{Y}_1}$, ..., $\mathbf{z}_{\mathcal{Y}_M}$, $w_1, ..., w_M$. According to PMO's reformulation (in Fig. 3), w_l is given by

$$w_l = \min_{\mathbf{z}_{\mathcal{X}_l} \geq \mathbf{0}} \{\mathbf{c}_{\mathcal{X}_l} \mathbf{z}_{\mathcal{X}_l} \,|\mathbf{A}_{\mathcal{X}_l} \mathbf{z}_{\mathcal{X}_l} \geq \mathbf{b}_{\mathcal{X}_l,\mathcal{Y}_l} - \mathbf{B}_{\mathcal{Y}_l} \mathbf{z}_{\mathcal{Y}_l}\}. \tag{13}$$

Since the MP doesn't know the optimal values of $\mathbf{z}_{\mathcal{X}_l}$, instead of using Eq. (13), it "guesses" the value of w_l based the *cut set* $\mathcal{H}$. In the subsequent **Stage 2**, each Sub_l verifies whether the "guessed" value of w_l is feasible and achieves the minimum data utility loss as defined in Eq. (13); if not, Sub_l proposes the addition of a new cut to be included in $\mathcal{H}$, thereby guiding the MP to refine w_l during the next iteration.

Initial cut. According to the **Property 3.2** of mDP constraints, the MP can initialize the cut set $\mathcal{H}$ by

$$\mathcal{H} = \left\{\mathbf{z}_i, \mathbf{z}_j | \mathbf{z}_i \leq e^{\epsilon D_{i,j}} \mathbf{z}_j, \; \forall r_i, r_j \in \cup_l \mathcal{Y}_l, (r_i, r_j) \notin \mathcal{E}\right\},$$

where $D_{i,j}$ is the shortest path distance between r_i and r_j in the mDP graph $\mathcal{G}$. In the following, we use $\{\overline{\mathbf{z}}_{\mathcal{Y}_1}, ..., \overline{\mathbf{z}}_{\mathcal{Y}_M}, \overline{w}_1, ..., \overline{w}_N\}$ to represent the optimal solution of the MP.

Stage 2: Subproblems. After the MP derives its optimal solution $\{\overline{\mathbf{z}}_{\mathcal{Y}_1}, ..., \overline{\mathbf{z}}_{\mathcal{Y}_M}, \overline{w}_1, ..., \overline{w}_N\}$ in **Stage 1**, each Sub_l validates whether $\overline{w}_l$ has achieved the minimum data utility loss,

$$\overline{w}_l = \min_{\mathbf{z}_{\mathcal{X}_l} \geq \mathbf{0}} \{\mathbf{c}_{\mathcal{X}_l} \mathbf{z}_{\mathcal{X}_l} \,|\mathbf{A}_{\mathcal{X}_l} \mathbf{z}_{\mathcal{X}_l} \geq \mathbf{b}_{\mathcal{X}_l,\mathcal{Y}_l} - \mathbf{B}_{\mathcal{Y}_l} \overline{\mathbf{z}}_{\mathcal{Y}_l}\}, \tag{14}$$

of which the *dual problem* can be formulated as the following LP problem:

$$\max \quad (\mathbf{b}_{\mathcal{X}_l,\mathcal{Y}_l} - \mathbf{B}_{\mathcal{Y}_l} \overline{\mathbf{z}}_{\mathcal{Y}_l})^\top \mathbf{a}_{\mathcal{X}_l} + \mathbf{c}_{\mathcal{Y}_l} \overline{\mathbf{z}}_{\mathcal{Y}_l} \tag{15}$$

$$\text{s.t.} \quad \mathbf{A}_{\mathcal{X}_l}^\top \mathbf{a}_{\mathcal{X}_l} \leq \mathbf{c}_{\mathcal{X}_l}, \; \mathbf{a}_{\mathcal{X}_l} \geq \mathbf{0}. \tag{16}$$

There are three cases of the dual problem:

C1: The optimal objective value is **unbounded**: By *weak duality* [Hillier, 2008], $\overline{\mathbf{z}}_{\mathcal{Y}_l}$ does not satisfy $\mathbf{A}_{\mathcal{X}_l} \mathbf{z}_{\mathcal{X}_l} \geq \mathbf{b}_{\mathcal{X}_l,\mathcal{Y}_l} - \mathbf{B}_{\mathcal{Y}_l} \overline{\mathbf{z}}_{\mathcal{Y}_l}$ for any $\mathbf{z}_{\mathcal{X}_l} \geq \mathbf{0}$. Since the dual problem is unbounded, there exists an *extreme ray* $\tilde{\mathbf{a}}_{\mathcal{X}_l}$ s.t. $\mathbf{A}_{\mathcal{X}_l}^\top \tilde{\mathbf{a}}_{\mathcal{X}_l} \leq \mathbf{0}$ and $(\mathbf{b}_{\mathcal{X}_l,\mathcal{Y}_l} - \mathbf{B}_{\mathcal{Y}_l} \overline{\mathbf{z}}_{\mathcal{Y}_l})^\top \tilde{\mathbf{a}}_{\mathcal{X}_l} > 0$. To ensure $\tilde{\mathbf{a}}_{\mathcal{X}_l}$ not to be an extreme ray in the next iteration, Sub_l suggests a *new cut* h (*feasibility cut*) to the MP:

$$h: \; (\mathbf{b}_{\mathcal{X}_l,\mathcal{Y}_l} - \mathbf{B}_{\mathcal{Y}_l} \mathbf{z}_{\mathcal{Y}_l})^\top \tilde{\mathbf{a}}_{\mathcal{X}_l} \leq \mathbf{0}. \tag{17}$$

C2: The optimal objective value is **bounded** with the solution $\overline{\mathbf{a}}_{\mathcal{X}_l}$: By *weak duality*, the optimal value of the dual

problem is equal to the optimal value of w_l constrained on the choice of $\overline{\mathbf{z}}_{\mathcal{Y}_l}$. In this case, Sub_l checks whether $\overline{w}_l < (\mathbf{b}_{\mathcal{X}_l,\mathcal{Y}_l} - \mathbf{B}_{\mathcal{Y}_l}\overline{\mathbf{z}}_{\mathcal{Y}_l})^\top \overline{\mathbf{a}}_{\mathcal{X}_l} + \mathbf{c}_{\mathcal{Y}_l}\overline{\mathbf{z}}_{\mathcal{Y}_l}$. If yes, then $\overline{w}_l < \min_{\mathbf{z}_{\mathcal{X}_l}\geq \mathbf{0}} \{\mathbf{c}_{\mathcal{X}_l}\mathbf{z}_{\mathcal{X}_l} \,|\mathbf{A}_{\mathcal{X}_l}\mathbf{z}_{\mathcal{X}_l} \geq \mathbf{b}_{\mathcal{X}_l,\mathcal{Y}_l} - \mathbf{B}_{\mathcal{Y}_l}\overline{\mathbf{z}}_{\mathcal{Y}_l}\}$, indicating that $\overline{w}_l$ guessed by the MP is lower than the minimum data utility loss. Therefore, Sub_l suggests a *new cut* $h:\ w_l \geq (\mathbf{b}_{\mathcal{X}_l,\mathcal{Y}_l} - \mathbf{B}_{\mathcal{Y}_l}\mathbf{z}_{\mathcal{Y}_l})^\top \overline{\mathbf{a}}_{\mathcal{X}_l} + \mathbf{c}_{\mathcal{Y}_l}\mathbf{z}_{\mathcal{Y}_l}$ to the MP to improve w_l in the next iteration.

C3: There is **no feasible solution**: By *weak duality*, the primal problem either has no feasible/unbounded solution. The algorithm terminates.

After adding the new cuts (from all the subproblems) to the cut set $\mathcal{H}$, the BD moves to the next iteration by recalculating the MP and obtaining updated $\overline{\mathbf{z}}_{\mathcal{Y}_l}$ and $\overline{w}_l$ $(l = 1, ..., M)$. As **Stage 1** and **Stage 2** are repeated over iterations, the MP collects more cuts from the subproblems, converging the solution $\overline{\mathbf{z}}_{\mathcal{Y}_l}$ and $\overline{w}_l$ to the optimal.

Proposition 3.3. *(*Upper and lower bounds of PMO's optimal*) [Rahmaniani* et al.*, 2017]*
(1) The optimal solution of the MP (Eq. (9) – (12)) offers a ***lower bound*** *of the optimal solution of the original PMO (Eq. (4)–(7)) (as the MP relaxes the constraints of PMO).*
(2) The solution of the subproblems (Eq. (15)-(16)), if it exists, combined with the solution $\overline{\mathbf{z}}_{\mathcal{Y}_l}$ of the MP, provides an ***upper bound*** *of the PMO's optimal (since their solutions form a feasible solution of the original PMO).*

Note that the optimal solution for PMO must lie within the gap between the upper bound and the lower bound in **Proposition 3.3**. The smaller the gap between these two bounds, the closer the solution of the BD is approaching the optimal solution of PMO. Fig. 15–Fig. 18 (in Section D.5 in Appendix) gives some examples of how the upper and lower bounds change over iterations. Considering a prolonged convergence tail, we termine the algorithm when the gap between the best upper and lower bounds falls below a predetermined threshold ξ (e.g., $\xi = 0.01$ in our experiments).

3.3 Time Complexity Analysis

In the two-stage framework of BD, the number of decision variables in the MP in **Stage 1** is $K(\sum_{l=1}^{M}|\mathcal{Y}_l|) + M$, including $K(\sum_{l=1}^{M}|\mathcal{Y}_l|)$ variables $(\mathbf{z}_{\mathcal{Y}1}, \ldots, \mathbf{z}_{\mathcal{Y}_M})$ to determine the perturbation vectors of the boundary records, and the M variables $(w_1, ..., w_M)$ to "guess" the utility loss of subproblems. Each Sub_l in **Stage 2** has $K|\mathcal{X}_l|$ decision variables to determine the perturbation vectors of the internal records $\mathbf{z}_{\mathcal{X}_l}$.

The MP and the subproblems are both formulated as LP problems, with time complexity depending on the number of decision variables [Cohen *et al.*, 2019]. To facilitate analysis, we use a monotonically increasing function $O(T(n))$ to represent the time complexity of LP given the number of decision variables n. In each iteration, the MP in Stage 1 has $O(T(K\sum_{l=1}^{M}|\mathcal{Y}_l|))$ time complexity, and each subproblem l in Stage 2 (run in parallel) has $O(T(K|\mathcal{X}_l|))$ time complexity. Since Stage 2 is terminated only after all subproblems are completed, its time complexity is $O(\max_l T(K|\mathcal{X}_l|)) = O(T(K\max_l|\mathcal{X}_l|))$ assuming all the subproblems are run in parallel. Hence, the total computation time of each iteration is given by $O(T(K\sum_{l=1}^{M}|\mathcal{Y}_l|) + T(K\max_l|\mathcal{X}_l|))$. Considering that both $\sum_{l=1}^{M}|\mathcal{Y}_l|$ and $\max_l|\mathcal{X}_l|$ is much smaller than the total number of records $|\mathcal{R}|$, the time complexity of each iteration of BD is significantly lower than that of the original PMO. There are two questions remain:

(1) *How many iterations are needed to converge the solution to the optimal?* In theory, if $|\cup_{l=1}^{M}\mathcal{Y}_l|$ is finite, generalized BD (including LP formulations) ends within a finite number of iterations for any given $\xi > 0$ [Floudas, 2009]. To assess practical applicability, we will examine the convergence of our specific BD framework in **Section 4.3** using different partitioning algorithms based on multiple datasets.

(2) *How to optimize the dataset partitioning to minimize the computation time $O(T(K\sum_{l=1}^{M}|\mathcal{Y}_l|) + T(K\max_l|\mathcal{X}_l|))$ in each iteration?* We introduce the detailed methods to address this problem in **Section 3.4**.

3.4 Secret Dataset Partitioning

According to the time complexity analysis of the BD framework in Section 3.3, we outline two primary objectives O-1 and O-2 for enhancing its computational efficiency when partitioning the secret dataset:
(O-1) *Maintain strong mDP constraints within each subset and weak mDP constraints across different subsets*. This not only facilitates relatively independent computation among subproblems but also reduces the number of decision variables (boundary records) in the MP.
(O-2) *Balance the number of records in different subsets*, to decrease the maximum time complexity of the subproblems.

Achieving the above two objectives is similar to solving the *RatioCut* problem [Hagen and Kahng, 1992], which seeks to partition a graph with minimal cuts while ensuring a well-balanced size of subgraphs, a problem known to be NP-hard.

Considering the computational trajectability, we apply a *Distance-Vector (DV)*-based dataset partitioning algorithm. As the mDP constraints of two perturbation vectors depend on their records' distance, we embed each record r_i by the *distance vector* $\mathbf{d}_i = [d_{i,1}, ..., d_{i,N}]$ to characterize its mDP relationship with all other records in the mDP graph. We consider that two records, r_i and r_j, exhibit a stronger connection when the Euclidean distance between their distance vectors, $\|\mathbf{d}_i - \mathbf{d}_j\|_2 = \sqrt{\sum_{l=1}^{N}(d_{i,l} - d_{j,l})^2}$, is lower, meaning that the two records are strongly coupled by the mDP constraints and also share similar mDP constraint relationship with other records. Accordingly, we partition the dataset to the subsets $\mathcal{R}_1, \ldots, \mathcal{R}_M$ using the distance vectors, which can be cast as the following *k-means clustering* formulation:

$$\min \ \sum_{l=1}^{M} \underbrace{\sum_{r_i\in\mathcal{R}_l}\|\mathbf{d}_i - \boldsymbol{\mu}_l\|_2}_{\text{Reflect the connection within } \mathcal{R}_l} \tag{18}$$

where $\boldsymbol{\mu}_l$ represents the *centroid* of the distance vectors in $\mathcal{R}_l$, i.e., $\boldsymbol{\mu}_l = \frac{\sum_{r_i\in\mathcal{R}_l}\mathbf{d}_i}{|\mathcal{R}_l|}$ $(l = 1, \ldots, M)$. Note that the objective function in Eq. (18) attains a lower value when the records within each partition are strongly connected and the sizes of $\mathcal{R}_1, ..., \mathcal{R}_M$ are well-balanced.

Besides the DV-based partitioning algorithm (labeled as "*k-mean-DV*" or "*k-m-DV*"), for comparison, we carried out three other benchmarks, *record-based partitioning*, *adjacency matrix-based partitioning*, and *balanced spectral clustering* [Hagen and Kahng, 1992], labeled as "*k-m-rec*", "*k-m-adj*", and "*BSC*", respectively. The details of the benchmarks are introduced in Section C in [Qiu, 2024].

4 Performance Evaluation

In this section, we test the performance of our approach using multiple datasets. We introduce the experiment settings in **Section 4.1**, and evaluate the performance of our method in terms of partitioning balance in **Section 4.2** and computation efficiency in **Section 4.3**. More comprehensive experimental results (including the comparison with exponential mechanisms and the performance evaluation with different parameters, etc.) can be found in **Section D** in [Qiu, 2024][1].

4.1 Settings

Datasets. As mDP has been primarily used in text embeddings and geo-location data [Imola *et al.*, 2022], we specifically choose text and geo-location datasets for the performance evaluation. Additionally, we assess our methods on a synthetic dataset.

(1) Geo-location dataset in the road network, composed of a set of "nodes" in the road network (retrieved by OpenStreetMap [OpenStreetMap, 2024]), including road intersections, forks, junctions where roads intersect with others, and points where the road changes direction. We selected the city *Rome, Italy* as the target region (specifically, the bounding area with coordinate $(lat = 41.66, lon = 12.24)$ as the south-west corner, and coordinate $(lat = 42.10, lon = 12.81)$ as the north-east corner). The target region encompasses a total of 43,160 discrete locations within Rome. In each experiment, we selected a relatively small target region in Rome, within which we sample 500 discrete locations in the road network. The distance metric between locations is defined by *Haversine distance*, i.e., the angular distance between two points on the surface of a sphere.

(2) Geo-location dataset in 10 grid maps, where each grid map is composed of 20×25 grid cells (the size of each cell is 1km×1km) in the target region, Rome, Italy. The center of each grid cell serves as a proxy for the cell's location. The distance metric between cells is defined as the *Euclidean distance* between their centers. In both datasets (1) and (2), we set $\eta = 2.00km$ and $\epsilon = 10.0/km$ by default.

(3) Text dataset, which comprises 2,000 words. Each word is represented by a 300-dimensional vector using MATLAB's `word2vec` function [MATLAB, 2024b] (pre-trained for 1 million English words), capturing both the semantic meaning of the word and its contextual usage. The distance between words is measured by the *Euclidean distance* between their vectors. **(4) Synthetic dataset**, which comprises 2,000 3-dimensional vectors, following the multivariate Gaussian distribution. The distance metric is defined as vectors' *Euclidean distance*. In both datasets (3)&(4), we set $\eta = 2.0, \epsilon = 10.0$ by default.

[1]The MATLAB source code of our method is available at: https://github.com/chenxiunt/MetricDP_BendersDecomposition

	Average size			
Dataset	**k-m-DV**	k-m-rec	k-m-adj	BSC
Road	**9.5±1.2**	10.1±0.8	17.7±1.3	15.4±0.8
Grid	**4.7±3.0**	3.7±6.9	14.9±1.7	3.0±4.7
Text	**12.6±2.6**	4.9±6.9	44.7±13.4	33.0±16.3
Syn	**9.6±1.9**	10.2±1.9	8.5±1.2	8.4±5.2
	Maximum size			
Dataset	**k-m-DV**	k-m-rec	k-m-adj	BSC
Road	**18.6±8.9**	16.1±10.1	34.6±12.5	32.1±15.3
Grid	**11.3±5.5**	10.2±7.6	28.3±11.0	17.2±10.3
Text	**31.8±6.4**	42.7±16.4	103.2±30.4	62.4±38.4
Syn	**17.9±2.9**	19.8±5.1	90.0±24.6	90.2±56.6

Table 1: Size of suproblems. Mean±1.96× standard deviation.

	Average size			
Dataset	**k-m-DV**	k-m-rec	k-m-adj	BSC
Road	**11.3±6.0**	37.9±24.7	40.4±34.3	64.7±14.5
Grid	**3.6±2.9**	8.8±4.3	9.0±60.9	18.6±3.3
Text	**8.0±4.4**	9.7±9.2	8.8±84.2	11.4±16.2
Syn	**10.9±4.3**	12.4±8.7	17.9±9.0	18.3±9.2
	Maximum size			
Dataset	**k-m-DV**	k-m-rec	k-m-adj	BSC
Road	**42.5±46.8**	98.8±44.4	118±59.2	121±79.1
Grid	**22.1±3.4**	10.3±5.1	99.5±95.1	27.1±4.2
Text	**36.0±15.4**	53.2±25.4	113±85.4	79.2±34.2
Syn	**35.9±45.5**	46.3±48.5	41.3±32.4	63.6±41.0

Table 2: Size of MP components. Mean±1.96× standard deviation.

Metrics: **(1) Problem sizes** of both subproblems and MPs. We measure the size of each decomposed problem by counting the number of perturbation vectors (each corresponds to a record) to derive. This metric reflects how effectively the dataset partitioning algorithms achieve balance in dividing the secret dataset. Note that the mDP graph of the MP may consist of multiple independent components. Hence, we measure the size of each component in the MP rather than the entire MP.

(2) Computation time to execute algorithms. The experiments are performed by a desktop with 13th Gen Intel Core i7 processor, 16 cores.

(3) Number of iterations for the BD to converge to the optiml solution. Considering the prolonged convergence tails of BD, we end the algorithm when the gap between its upper and lower bounds is lower than the threshold $\xi = 0.01$.

4.2 Secret Dataset Partitioning

Table 1 and Table 2 compare the sizes of subproblems and MP's components generated by the dataset partitioning algorithms, "k-m-DV", "k-m-rec", "k-m-adj", and "BSC". Each method undergoes testing on 10 instances for every dataset. The tables present the mean value±1.96×standard deviation of the average/maximum size of subproblems and MP components across different instances. The detailed visual representation of the dataset partitioning in different datasets is given in Fig. 11–Fig. 14 in [Qiu, 2024].

Datasets	LP-based methods					
	Benders decomposition				Classic LP	
	k-m-DV	k-m-rec	k-m-adj	BSC	Dual-simplex	Interior-point
Geo-location dataset (road network)	**290.7±33.5**	1066.9±154.4	max iter. ex.	max iter. ex.	max iter. ex.	max iter. ex.
Geo-location dataset (grid maps)	**161.4±14.3**	159.8±4.8	max iter. ex.	238.2±6.0	max iter. ex.	max iter. ex.
Text dataset	**27.2±0.3**	925.0±79.0	58.7±9.7	1218.6±78.3	max iter. ex.	max iter. ex.
Synthetic dataset	**185.1±23.9**	162.7±12.0	max iter. ex.	434.1±21.1	max iter. ex.	max iter. ex.

Table 3: Computation time (seconds). Mean±1.96× standard deviation. "max iter. ex." means "maximum iterations exceeded"

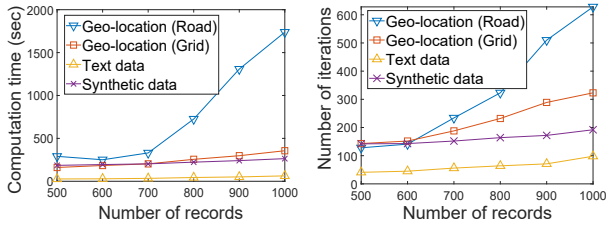
The tables show that, in all four datasets, k-m-DV partitions the PMO (containing 500 secret records) into sufficiently small subproblems and MP components, with the mean maximum size not exceeding 50. This decomposition facilitates efficient resolution of the individual problems using classical LP solvers such as the `dual-simplex` method [Hillier, 2008]. In contrast to k-m-DV, the size variation in the decomposed problems is higher for the other three methods. Specifically, the maximum problem size of the decomposed problems for k-m-adj and BSC is higher than 100, indicating a challenge in solving these larger problems efficiently. Both k-m-adj and BSC are less effective in balancing the size of decomposed problems, as they focus on the mDP constraint between neighboring records without considering overall constraint distribution. While k-m-rec demonstrates comparable performance to k-m-DV when applied to the geo-location data and synthetic data, it falls short in achieving balanced partitioning for text data, due to its limited ability to capture mDP constraints in high-dimensional (300-dimensional) spaces.

4.3 Computational Efficiency

Table 3 compares the computation time of the BD framework using the four dataset partitioning algorithms, "k-m-DV", "k-m-rec", "k-m-adj", and "BSC", and two classic LP solvers, `dual-simplex` and `interior-point`, both of which are provided by the MATLAB LP toolbox `linprog` [MATLAB, 2024a]. A detailed comparison of the BD convergence of the four partitioning algorithms using the four datasets is shown in Fig. 15–Fig. 18 in [Qiu, 2024].

As shown in Table 3, both classic LP methods stop prematurely without achieving the optimal solution as they exceed the maximum number of iterations (the parameter `MaxIter` is set by 10,000). Within the BD framework, on average, k-m-DV achieves the lowest computation time. This efficiency can be attributed to the balanced sizes of decomposed subproblems, as shown in Table 1 and Table 2. As discussed in Section 3.3, the computation time of BD in each iteration is significantly influenced by the computation time of the largest subproblem, which is relatively smaller when the size variation of decomposed problems is minimized. Additionally, a smaller subproblem, on average, needs fewer iterations to converge to a feasible solution within the BD framework (refer to Fig. 6 in [Qiu, 2024], which demonstrates a positive correlation between the subproblem size and the number of iterations to find a feasible solution).

Finally, we expand the size of the secret dataset from 500 records to 1,000 records and assess the computational efficiency and number of iterations of "k-m-DV" in Fig. 4(a)(b),



(a) Average computation time (b) Avg. number of iterations

Figure 4: Computation efficiency of k-m-DV in larger scale datasets.

respectively. In alignment with the recent related work [Imola *et al.*, 2022], we allocate a calculation time limit of up to 1,800 seconds. Notably, the computation time for "k-m-DV" remains below 6 minutes when using grid maps, text data, and synthetic datasets. However, when using the road map location dataset, the computation time experiences a significant increase, reaching approximately 1,800 seconds when the dataset size is 1,000. This difference can be attributed to the uneven distribution of nodes in the large-scale road network, with low node density in suburban areas and high node density in downtown areas. This leads to some larger decomposed problems, resulting in a longer time to solve.

5 Conclusion

In this paper, we propose improving the scalability of LP-based mDP through secret dataset partitioning and BD. Experimental results with diverse datasets show that our approach expands the scale of existing LP-based mDP by approximately 9 times. Additionally, our findings highlight the importance of balancing the size between decomposed subproblems and MP components when implementing BD.

We identify several promising directions for further improvement. First, while BD enhances metric-DP scalability, its current convergence (up to 650 iterations) is challenging for time-sensitive applications. Leveraging *reinforcement learning (RL)* could accelerate BD convergence by treating cut selection in Stage 2 as a parameterized stochastic policy. A trained RL model can identify a sequence of cuts for fixed subproblem coefficients, eliminating the need for re-training with each new problem instance. Second, recognizing that when the size of secret datasets is higher (e.g., over 1,000), the decomposed MP might remain too large for current LP solvers to handle. A potential direction is to explore a combination of multiple decomposition techniques, such as Danzig-Wolfe decomposition, based on the mDP constraint features.

Acknowledgments

This research is partially supported by U.S. NSF grants CNS2136948 and CNS-2313866.

References

[Andrés *et al.*, 2013] Miguel E. Andrés, Nicolás E. Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Geo-indistinguishability: differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC conference on Computer amp; communications security - CCS '13*. ACM Press, 2013.

[Bordenabe *et al.*, 2014] Nicolás E. Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Optimal geo-indistinguishable mechanisms for location privacy. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, page 251–262, New York, NY, USA, 2014. Association for Computing Machinery.

[Chatzikokolakis *et al.*, 2013] Konstantinos Chatzikokolakis, Miguel E. Andrés, Nicolás Emilio Bordenabe, and Catuscia Palamidessi. Broadening the scope of differential privacy using metrics. In Emiliano De Cristofaro and Matthew Wright, editors, *Proc. of Privacy Enhancing Technologies*, pages 82–102, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[Chatzikokolakis *et al.*, 2015] Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Marco Stronati. Constructing elastic distinguishability metrics for location privacy. *Privacy Enhancing Technologies (PoPETs)*, 2015:156–170, 2015.

[Cohen *et al.*, 2019] Michael B. Cohen, Yin Tat Lee, and Zhao Song. Solving linear programs in the current matrix multiplication time. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 938–942, New York, NY, USA, 2019. Association for Computing Machinery.

[Dwork *et al.*, 2006] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, pages 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[Fawaz and Shin, 2014] Kassem Fawaz and Kang G. Shin. Location privacy protection for smartphone users. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, page 239–250, New York, NY, USA, 2014. Association for Computing Machinery.

[Fernandes *et al.*, 2019] Natasha Fernandes, Mark Dras, and Annabelle McIver. Generalised differential privacy for text document processing. In Flemming Nielson and David Sands, editors, *Proc. of Principles of Security and Trust*, pages 123–148, Cham, 2019. Springer International Publishing.

[Feyisetan and Kasiviswanathan, 2021] Oluwaseyi Feyisetan and Shiva Kasiviswanathan. Private release of text embedding vectors. In *Proc. of the First Workshop on Trustworthy Natural Language Processing*, pages 15–27, Online, June 2021. Association for Computational Linguistics.

[Feyisetan *et al.*, 2019] Oluwaseyi Feyisetan, Tom Diethe, and Thomas Drake. Leveraging hierarchical representations for preserving privacy and utility in text. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 210–219, Los Alamitos, CA, USA, nov 2019. IEEE Computer Society.

[Floudas, 2009] Christodoulos A. Floudas. *Generalized benders decompositionGeneralized Benders Decomposition*, pages 1162–1175. Springer US, Boston, MA, 2009.

[Hagen and Kahng, 1992] Lars Hagen and Andrew B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 11(9):1074–1085, 1992.

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

[Imola *et al.*, 2022] Jacob Imola, Shiva Kasiviswanathan, Stephen White, Abhinav Aggarwal, and Nathanael Teissier. Balancing utility and scalability in metric differential privacy. In *Proc. of UAI 2022*, 2022.

[Lütkepohl, 1996] Helmut Lütkepohl. *Handbook of Matrices*. Graduate texts in mathematics. Springer, 1996.

[MATLAB, 2024a] MATLAB. linprog: Solve linear programming problems. https://www.mathworks.com/help/optim/ug/linprog.html, 2024. Accessed in January 2024.

[MATLAB, 2024b] MATLAB. word2vec: Map word to embedding vector. https://www.mathworks.com/help/textanalytics/ref/wordembedding.word2vec.html, 2024. Accessed in January 2024.

[McSherry and Talwar, 2007] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 94–103, 2007.

[Ng *et al.*, 2001] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001.

[OpenStreetMap, 2024] OpenStreetMap. Openstreetmap homepage. https://www.openstreetmap.org/, 2024. Accessed: 2020-04-07.

[Palomar and Chiang, 2006] Daniel P. Palomar and Mung Chiang. A tutorial on decomposition methods for network utility maximization. *IEEE Journal on Selected Areas in Communications*, 24(8):1439–1451, 2006.

[Pappachan *et al.*, 2023] Primal Pappachan, Chenxi Qiu, Anna Squicciarini, and Vishnu Sharma Hunsur Manjunath. User customizable and robust geo-indistinguishability for location privacy. In *Proc. of International Conference on Extending Database Technology (EDBT)*, 2023.

[Qiu and Squicciarini, 2019] Chenxi Qiu and Anna C. Squicciarini. Location privacy protection in vehicle-based spatial crowdsourcing via geo-indistinguishability. In *Proc. of IEEE ICDCS*, pages 1061–1071, 2019.

[Qiu *et al.*, 2020] Chenxi Qiu, Anna Squicciarini, Zhuozhao Li, Ce Pang, and Li Yan. Time-efficient geo-obfuscation to protect worker location privacy over road networks in spatial crowdsourcing. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 1275–1284, New York, NY, USA, 2020. Association for Computing Machinery.

[Qiu *et al.*, 2022a] Chenxi Qiu, Anna C. Squicciarini, Ce Pang, Ning Wang, and Ben Wu. Location privacy protection in vehicle-based spatial crowdsourcing via geo-indistinguishability. *IEEE Transactions on Mobile Computing*, pages 1–1, 2022.

[Qiu *et al.*, 2022b] Chenxi Qiu, Li Yan, Anna Squicciarini, Juanjuan Zhao, Chengzhong Xu, and Primal Pappachan. Trafficadaptor: an adaptive obfuscation strategy for vehicle location privacy against traffic flow aware attacks. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '22, New York, NY, USA, 2022. Association for Computing Machinery.

[Qiu, 2024] Chenxi Qiu. Enhancing scalability of metric differential privacy via secret dataset partitioning and benders decomposition. https://arxiv.org/abs/2405.04344, 2024. Accessed in June 2024.

[Rahmaniani *et al.*, 2017] Ragheb Rahmaniani, Teodor Gabriel Crainic, Michel Gendreau, and Walter Rei. The benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259(3):801–817, 2017.

[Shokri *et al.*, 2012] Reza Shokri, George Theodorakopoulos, Carmela Troncoso, Jean-Pierre Hubaux, and Jean-Yves Le Boudec. Protecting location privacy: optimal strategy against localization attacks. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, page 617–627, New York, NY, USA, 2012. Association for Computing Machinery.

[Stroock, 2010] Daniel W. Stroock. *Probability Theory: An Analytic View*. Cambridge University Press, 2nd edition, 2010.

[von Luxburg, 2007] Ulrike von Luxburg. A tutorial on spectral clustering. https://arxiv.org/abs/0711.0189, 2007. Accessed in June 2024.

[Wang *et al.*, 2016] Leye Wang, Daqing Zhang, Dingqi Yang, Brian Y. Lim, and Xiaojuan Ma. Differential location privacy for sparse mobile crowdsensing. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 1257–1262, 2016.

[Wang *et al.*, 2017] Leye Wang, Dingqi Yang, Xiao Han, Tianben Wang, Daqing Zhang, and Xiaojuan Ma. Location privacy-preserving task allocation for mobile crowdsensing with differential geo-obfuscation. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 627–636, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee.

[Wilson, 1986] Robin J Wilson. *Introduction to Graph Theory*. John Wiley & Sons, Inc., USA, 1986.

[Yu *et al.*, 2017] Lei Yu, Ling Liu, and Calton Pu. Dynamic differential location privacy with personalized error bounds. In *Proc. of IEEE NDSS*, 2017.