Differential Privacy Meets Maximum-weight Matching

Panayiotis Danassis, Aleksei Triastcyn, Boi Faltings
Artificial Intelligence Laboratory
École Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland
{firstname.lastname}@epfl.ch

Abstract

When it comes to large-scale multi-agent systems with a diverse set of agents, traditional differential privacy (DP) mechanisms are ill-matched because they consider a very broad class of adversaries, and they protect all users, independent of their characteristics, by the same guarantee. Achieving a meaningful privacy leads to pronounced reduction in solution quality. Such assumptions are unnecessary in many real-world applications for three key reasons: (i) users might be willing to disclose less sensitive information (e.g., city of residence, but not exact location), (ii) the attacker might possess auxiliary information (e.g., city of residence in a mobility-on-demand system, or reviewer expertise in a paper assignment problem), and (iii) domain characteristics might exclude a subset of solutions (an expert on auctions would not be assigned to review a robotics paper, thus there is no need for indistinguishability between reviewers on different fields).

We introduce Piecewise Local Differential Privacy (PLDP), a privacy model designed to protect the utility function in applications where the attacker possesses additional information on the characteristics of the utility space. PLDP enables a high degree of privacy, while being applicable to real-world, unboundedly large settings. Moreover, we propose PALMA, a privacy-preserving heuristic for maximum-weight matching. We evaluate PALMA in a vehicle-passenger matching scenario using real data and demonstrate that it provides strong privacy, a median of $\varepsilon = 0.44$, and high quality matchings ($10.8\%$ worse than the non-private optimal).

1 Introduction

One of the fundamental problems in multi-agent systems is finding an optimal allocation, i.e., solving a maximum-weight matching (MWM) problem. A wide range of applications – spanning from mobility-on-demand systems and ridesharing (Danassis et al. 2019), to kidney exchange (Roth, Sönmez, and Unver 2005) – can be formulated and solved as a maximum-weight matching problem. Real-world matching problems pose three significant challenges: (i) they may occur in unboundedly large settings (e.g., resource allocation in urban environments), (ii) they are distributed and information-restrictive (agents have partial observability and inter-agent communication might not be available (Stone et al. 2010)), and finally, (iii) individuals have to reveal their preferences in order to get a high quality match, which brings forth significant privacy risks.

In this work, we propose PALMA (Privacy-preserving ALtruistic MAtching), a matching heuristic designed to tackle all of the aforementioned challenges.

PALMA is a privacy-preserving adaptation of ALMA (Danassis, Filos-Ratsikas, and Faltings 2019), a recently proposed heuristic for real-world, large-scale, applications. ALMA is decentralized, requires no communication between the participants, and converges in constant (to the total problem size) time – in the realistic case where each agent is interested in a (fixed size) subset of the total resources.

The third challenge requires protecting the utility functions of the agents. In recent years, Differential Privacy (DP) (Dwork 2006) has emerged as the de facto standard for protecting the privacy of individuals. Yet, conventional DP often requires adding a lot of random noise to achieve a meaningful guarantee, which in turn leads to pronounced drop in the solution quality. More often than not, this is not due to the inherent difficulty of the problem at hand, but rather due to the generality of the DP definition. Not only does DP consider a very broad class of adversaries, it also does not make any assumptions on the data it protects. While this property is being praised as one of the strongest arguments in favor of DP, it can be completely redundant in many real-world applications. For instance, in a ridesharing application, it is most likely acceptable to disclose the fact that an individual is in London rather than New York. However, disclosing the precise location within London is undesirable. Similarly, in a paper assignment problem (reviewers to manuscripts), ensuring indistinguishably between an expert on Markets & Auctions, and one on Robotics might be futile, especially if the attacker possesses additional information (e.g., the tracks of the papers under review) that would exclude infeasible matches.

In this paper, we consider the problem of hiding the utility function, and we motivate and develop an ‘application-aware’ threat model and privacy definition (Piecewise Local Differential Privacy – PLDP) which takes into account the ‘distance’ between the images of two utility functions. The level of protection depends on that distance; agents with utility functions that have images close in distance to each other would be indistinguishable from the attacker’s point of view. The definition is inspired by existing work on ‘data-aware’ privacy notions (Triastcyn and Faltings).
We introduce Piecewise Local Differential Privacy (PLDP), a variant of differential privacy designed to protect the utility function in multi-agent applications where the attacker possesses additional information on the characteristics of the utility space. This novel approach to quantifying privacy enables significant improvements in solutions quality and strong theoretical privacy guarantees, while being applicable in real-world, unboundedly large settings.

(1) We propose PALMA, a privacy-preserving heuristic for maximum-weight matching in real-world, large-scale applications, which combines ALMA (Danassis, Filos-Ratsikas, and Faltings 2019), PLDP, and a privacy accounting method for iterative algorithms.

(2) We evaluate PALMA in a realistic taxi-passenger matching scenario, using real data from NYC TLC 2016. PALMA is able to provide a high degree of privacy, $\varepsilon \leq 3$ and a median value of 0.44 for $\delta = 1e^{-5}$, and matchings of high quality (only 10.8% worse social welfare compared to the optimal, non-private solution).

1.1 Our Contributions

We outline the differences in Section 2. Finally, we combine PLDP with ALMA to create a decentralized, privacy-preserving heuristic (PALMA) for large-scale maximum-weight matching problems.

2 Related Work

Finding a maximum weight matching in a weighted graph is one of the best-studied combinatorial optimization problems in the literature (see Su 2015, Lovász and Plummer 2009). There is a plethora of polynomial time algorithms, with the Hungarian (Kuhn 1955) and the blossom (Edmonds 1965) algorithms being the most prominent central ones for the bipartite and general variant, respectively. In real-world problems, a centralized coordinator is not always available, and if so, it has to know the utilities of all the participants which is often not feasible and poses significant privacy risks. Decentralized algorithms (e.g., Giordani, Lujak, and Martelli 2010, Ismail and Sun 2017, Zavlanos, Spesivtsev, and Pappas 2008) solve the former problem, yet they require polynomial computational time and polynomial number of messages. Thus, while the problem has been solved from an algorithmic perspective – having both centralized and decentralized polynomial algorithms – it is not so from the perspective of multi-agent systems, for three key reasons: (i) complexity, (ii) communication, and (iii) privacy.

The proliferation of intelligent systems will give rise to large-scale, multi-agent based technologies. Algorithms for maximum-weight matching, whether centralized or distributed, have runtime that increases with the total problem size, even in the realistic case where agents are interested in a small number of resources. Thus, they can only handle problems of bounded size. Moreover, they require a significant amount of inter-agent communication. Yet, communication might not always be an option (Stone et al. 2010), and sharing utility tables, plans, and preferences creates high overhead. ALMA on the other hand achieves constant in the total problem size running time – under reasonable assumptions on the preference domain of the agents – while requiring no message exchange (i.e., no communication network) between the participating agents (Danassis, Filos-Ratsikas, and Faltings 2019). The proposed approach, PALMA, preserves the aforementioned two properties of ALMA, thus dealing with the first two of the posed challenges.

Differential Privacy (DP) (Dwork 2006, Dwork et al. 2006a,b) has emerged as the de facto standard for protecting the privacy of individuals. Informally, DP captures the increased risk to an individual’s privacy incurred by his participation. A variation of differential privacy, especially useful in our context given the decentralized nature of PALMA, is local differential privacy (LDP) (Dwork, Roth et al. 2014). LDP is a generalization of DP that provides a bound on outcome probabilities for any pair of individual agents rather than populations differing on a single agent. Intuitively, it means that one cannot hide in the crowd. Another strength of LDP is that it does not use a centralized model to add noise—individuals sanitize their data themselves—providing privacy protection against a malicious data curator. As a result, LDP requires adding even more random noise to achieve a meaningful bound, which would result in decline of solution quality. (L)DP ignores specifics of AI applications, such as a focus on a given task or a particular data distribution.

Inspired by the notions of Bayesian DP (Triastcyn and Faltings 2019) – which is based on the observation that machine learning models are designed and tuned for a particular data distribution (e.g., an MRI dataset is very unlikely to contain a picture of a car) and such prior distribution of data is also often available to the attacker – and metric-based DP (Chatzikokolakis et al. 2013) and Geo-indistinguishability (Andrés et al. 2013) – where indistinguishability depends on an arbitrary notion of distance – we propose a new privacy model, namely Piecewise Local Differential Privacy (PLDP). PLDP takes into account the ‘distance’ between the images of two utility functions, and the level of protection depends on that distance. The rationale is that instead of guaranteeing local privacy in the entire domain of agents, which can be quite difficult and would result in low quality solutions due to excessive noise, we focus on indistinguishability of agents with similar preferences.

2 Piecewise Local Differential Privacy

In this section, we provide a detailed description of our privacy model, named Piecewise Local Differential Privacy (PLDP).

2.1 Definition

We consider a randomized function $M: D \rightarrow A$ with domain $D$ and range $A$. In the context of assignment problems in multi-agent systems, $D$ is the space of all utility functions...
and $A$ is the action space. In the context of PALMA specifically, an action is either an attempt to acquire a certain resource, or a back-off from a previously contested resource, as will explained in the following section.

**Definition 1.** Let $\varphi(\cdot)$ be a set function that fragments $D$ into a collection of subsets $\{D_i\}$. Then, a randomized algorithm $M : D \rightarrow A$ satisfies $(\varepsilon, \delta, \varphi)$-piecewise local privacy if for any two inputs $x, x' \in D_i, \forall i$, and for any set of outcomes $S \subset A$ the following holds:

$$
\Pr [M(x) \in S] \leq e^\varepsilon \Pr [M(x') \in S] + \delta.
$$

The rationale behind this notion is the following. Instead of guaranteeing local privacy in the entire domain of agents, which may be quite difficult, we focus on indistinguishability of agents with similar preferences. We fragment the space of utilities into regions and guarantee privacy within these regions but not between them.

A useful real-world analogy is ZIP codes. Assume we would like to release some location statistic with PLDP and we choose $\varphi$ such that the initial location space is mapped into ZIP codes. Then, $(\varepsilon, \delta, \varphi)$-PLDP guarantee would certify that the reported statistic is $(\varepsilon, \delta)$-locally private within each ZIP code. However, it would not tell us anything about privacy of the reported statistic outside the given ZIP code.

Note that PLDP is a straightforward relaxation of local privacy and all the properties of LDP are satisfied within sub-domains $D_i$. Moreover, in the example above, piecewise local privacy resembles another well-known privacy notion, geo-indistinguishability (Andrés et al. 2013), which is based on a generalization of DP (Chatzikokolakis et al. 2013). Despite the similarities, there is a notable distinction: regions within which privacy is protected are predefined by $\varphi$, instead of being centered at $x$. This allows to use tighter composition theorems developed for the conventional DP, which gives significant advantage in real-world settings by reducing the growth of $\varepsilon$ from linear w.r.t. the total number of algorithm iterations $T$ to $O(\sqrt{T})$ (Abadi et al. 2016). However, the downside is that the privacy guarantee is limited to the given region rather than fading gradually with increasing region radius.

### 3 PALMA: A Privacy-Preserving Maximum-Weight Matching Heuristic

In this section we introduce PALMA (Privacy-preserving ALtruistic MAtrching), a privacy-preserving adaptation of ALMA (Danassis, Filos-Ratsikas, and Faltings 2019). We start by describing the problem of finding a maximum-weight matching. For simplicity, we will focus on bipartite graphs (i.e., the assignment problem), but (P)ALMA can be applied in general graphs as well (see Danassis et al. 2019 for an example application of ALMA in general graphs). Finally, we describe the employed privacy mechanisms, and the privacy accounting method.

#### 3.1 The Assignment Problem

The assignment problem consists of finding a maximum weight matching in a weighted bipartite graph, $\mathcal{G} = (\mathcal{N} \cup \mathcal{R}, \mathcal{E})$. In the studied scenario, $\mathcal{N} = \{1, \ldots, N\}$ agents compete to acquire $\mathcal{R} = \{1, \ldots, R\}$ resources. The weight of an edge $(n, r) \in \mathcal{E}$ represents the utility $u_n(r) \in [0, 1]$ agent $n$ receives by acquiring resource $r$. Each agent can acquire at most one resource, and each resource can be assigned to at most one agent. The goal is to maximize the social welfare (sum of utilities), i.e.,

$$
\max_{\mathbf{x} \in \{0, 1\}^R} \sum_{(n, r)} u_n(r) x_{n,r} - \text{where the variable } x_{n,r} = 1 \text{ if the edge is contained in the matching and 0 otherwise,}
$$

and $\mathbf{x} = (x_1, \ldots, x_{N,R}) - subject to \sum_r x_{n,r} = 1, \forall n \in \mathcal{N}$, and $\sum_n x_{n,r} = 1, \forall r \in \mathcal{R}$.

For simplicity, in the rest of the paper we assume $N = R$. This is not required by (P)ALMA. If $R > N$ some resources will remain free, while if $N > R$ some agents will fail to acquire a resource (convergence in the latter case implies that the state of the agent does not change).

#### 3.2 Learning Rule

PALMA is run independently and in parallel by all the agents. An agent running PALMA converges to a resource through repeated trials. We assume that agents can observe feedback from their environment to inform collisions and detect free resources. The pseudo-code of PALMA can be found in Algorithm 1.

We assume that each agent is interested in (potentially) a subset of the total resources $\mathcal{Q} \subseteq \mathcal{R}$. Let $\mathcal{A} = \{Y, A_1, \ldots, A_{|Q|}\}$ denote the set of actions, where $Y$ refers to yielding, and $A_r$ refers to accessing resource $r$, and let $g$ denote the agent’s strategy. As long as an agent has not acquired a resource yet, at every time-step, there are two possible scenarios: If $g = A_r$, (strategy points to resource $r$), then agent $n$ attempts to acquire that resource. If there is a collision, the colliding parties back-off with some probability $P_B^g(\cdot)$. Otherwise, if $g = Y$, the agent choses a resource $r$ for monitoring according to probability $P_B^y(\cdot)$. If the resource is free, he sets $g \leftarrow A_r$.

**Resource Selection Distribution**

In the original implementation of ALMA, each agent sorts the resources in decreasing order of utility $(r_1, r_2, \ldots, r_x, \ldots, r_R)$, then, moves in a sequential manner, starting from the most preferred resource $(r_1)$, and moving down the list until he acquires one. This method of resource selection results in the highest social welfare (sum of utilities), but it is impossible to guarantee privacy due to the deterministic nature of the selection process. On the other end of the spectrum, we can select a resource in a weighted random fashion, where resource $r_i$ is selected with probability $\frac{u_{n_i}(r_i)}{\sum_{r \in R} u_{n_i}(r)}$. This method provides high degree of privacy, but can result in low social welfare. To elaborate the latter, consider the following adversarial scenario: in a large-scale urban domain $(|R| \rightarrow \infty)$ where agents are interested only in resources that are physically close to them, the majority of resources would have utility $\approx 0$. If we select a resource in a weighted

---

2Feedback can be received for example by the use of sensors, or by a single bit (0/1) feedback from the resource (note that these messages would be between the requesting agent and the resource, not between the participating agents themselves).
at random fashion, the probability of selecting a low utility resource would be high – due to the large number of resources – resulting in low social welfare.

In this work, we combine the aforementioned two approaches. Let $N^n$ denote the set of all agents that are in the same region of utility space as $n$, i.e., $N^n = \{n' : \varphi(u_n(\cdot)) = \varphi(u_{n'}(\cdot))\}$. We refer to $N^n$ as the set of neighbors of $n$. Note that the neighbors of an agent do not need to be in $N$, we account for every potential agent (i.e., $\cup_{n' \in N} N^n \supset N$). The neighbors are the set of agents that PLDP guarantees indistinguishability. Then, agent $n$ generates the sets $(\mathcal{R}_n^1, \mathcal{R}_n^2 \ldots, \mathcal{R}_n^x, \ldots, \mathcal{R}_n^h)$, where the set $\mathcal{R}_n^x$ contains the $x^\text{th}$ most preferred resource of each neighbor, i.e., $\mathcal{R}_n^x = \cup_{r \in N^n} \{r_n^x\}$.

Agent $n$ moves in a sequential manner from set to set (starting from the set of the most preferred resources, $\mathcal{R}_n^1$, and looping back to it after $\mathcal{R}_n^h$). The resource selection is performed in a weighted at random fashion in the sets $\mathcal{R}_n^x$. Specifically, at step $s \in \{1, \ldots, R\}$, agent $n$ will select resource $r_i \in \mathcal{R}_n^x$ with probability given by:

$$P_B^n(i, s, \zeta_s) = (1 - \zeta_s) \frac{u_n(r_i)}{\sum_{r \in \mathcal{R}_n^x} u_n(r)} + \frac{\zeta_s}{|\mathcal{R}_n^x|}$$

(1)

where $\zeta_s$ tunes the magnitude of the introduced randomness.

**Back-off Distribution** The back-off probability, $P_B^n(\cdot) : \mathcal{R} \rightarrow [0, 1]$, is computed individually and locally based on each agent’s expected utility loss of switching:

$$loss_n(i, s) = u_n(r_i) - \sum_{r_j \in \mathcal{R}_{n+s+1}} \frac{u_n(r_j)}{\sum_{r \in \mathcal{R}_{n+s+1}} u_n(r)} u_n(r_j)$$

(2)

The actual back-off probability can be computed with any monotonically decreasing function $f$ on $loss_n(\cdot)$, e.g.:

$$f(loss) = \begin{cases} 
1 - \gamma, & \text{if } loss \leq \gamma \\
\gamma, & \text{if } 1 - loss \leq \gamma \\
1 - loss, & \text{otherwise}
\end{cases}$$

(3)

were $\gamma$ places a threshold on the minimum / maximum back-off probability. Finally, $P_B^n(\cdot)$ is given by Eq. (4), where $\zeta_B$ tunes the magnitude of the introduced randomness.

$$P_B^n(i, \zeta_B) = (1 - \zeta_B)f(loss_n(\cdot)) + \frac{\zeta_B}{2}$$

(4)

According to the above distribution, agents that do not have good alternatives will be less likely to back-off and vice versa. The ones that do back-off select an alternative resource, according to the resource selection probability distribution $P_B^n(\cdot)$, and examine its availability.

**Bounding the Set of Desirable Resources** An important characteristic of many real-world applications is that there is typically a cost associated with acquiring a resource. As a result, each agent is typically interested in a subset of the total resources, i.e., $Q^n \subset \mathcal{R}$. For example, a taxi driver would not be willing to drive to the other end of the city to pick up a low fare passenger, a driver would not be willing to charge his vehicle at a station in a different part of the city, and a reviewer would not be willing to review a paper outside his scope of expertise. This results in faster convergence, but can also potentially lead to higher social welfare. The sets ($\mathcal{R}_1^1, \mathcal{R}_2^x, \ldots, \mathcal{R}_x^h$) can be contracted in the same manner as before.

**Convergence** Theorem 2.1 of [Danassis, Filos-Ratsikas, and Faltings 2019] proves that PALMA converges in polynomial time. In fact, under the aforementioned assumption that each agent is interested in a subset of the total resources (i.e., $Q^n \subset \mathcal{R}$) and thus at each resource there is a bounded number of competing agents ($|V|^r \subset |N|$) Corollary 2.1.1 of [Danassis, Filos-Ratsikas, and Faltings 2019] proves that the expected number of steps any individual agent requires to converge is independent of the total problem size (i.e., $N$ and $R$). In other words, by bounding these two quantities (i.e., we consider $|Q^n|, |V|^r$ to be constant functions of $N$, $R$), the convergence time is constant in the total problem size $N, R$.

### 3.3 Privacy Mechanisms

PALMA implements two separate defense mechanisms, applied in three different parts of the algorithm, all of which can be separately tuned depending on the desired level of privacy at each stage.

---

Table 1: **Nomenclature, Algorithm**

| $s$ | Current step (indicates a specific set $\mathcal{R}_s^x$) |
| $g$ | Specifies which resource to access |
| $\{Y, A_1, \ldots, A_n\}$ | $A_r$ refers to accessing resource $r$ |
| $P_B^n() : \mathcal{R} \rightarrow [0, 1]$ | Resource selection probability distribution |
| $P_B^n() : \mathcal{R} \rightarrow [0, 1]$ | Back-off probability distribution |

---

Algorithm 1 **PALMA**: Privacy-preserving ALtruistic MAtrix Matching heuristic.

1: Initialize $s ← 1$
2: Initialize $g \sim P_B^n(\cdot)$
3: **procedure** PALMA(subsampled)
4: if subsampled then
5: if $g = A_r$ then
6: Agent $n$ attempts to acquire $r$
7: if Collision($r$) then
8: Back-off (set $g ← Y$) with prob. $P_B^n(\cdot)$
9: else ($g = Y$)
10: $s ← (s + 1) \bmod R$
11: Agent $n$ monitors $r \sim P_B^n(\cdot)$
12: if Free($r$) then set $g ← A_r$
13: return $r$, such that $g = A_r$
• **Randomized Action Selection** The first mechanism is based on the idea of randomized response (Warner 1965), and involves adding controlled randomness in (i) the resource selection and (ii) back-offs, parametrized by ζS and ζB, respectively (see Eq. 1 and 3). The idea is that the agent first flips a coin to decide whether to act truthfully. Then, with probability 1 − ζS (or 1 − ζB), the agent plays according to its true selection (or back-off) function; with probability ζS (or ζB), the agent plays uniformly at random.

Lastly, each agent has a privacy budget of ε = ζ + Bn. Upon depletion in the course of using the above mechanisms, the agent will play random actions.

4 Privacy Accounting

Since PALMA is an iterative algorithm, we need to compute (ε, δ) guarantees over multiple applications of the privacy mechanism. This can be done via privacy accounting methods. The most simple of these methods is to use the basic composition theorem of differential privacy (Dwork, Roth 2017). More advanced tools have been developed in the machine learning community (Abadi et al. 2016; Mironov 2017), but they have largely focused on the Gaussian noise mechanism in particular.

We employ the accounting framework introduced by Triastcyn and Faltings (2020) and extended to generic subsampled mechanisms (Triastcyn 2020). While developed for the notion of Bayesian DP, this framework is applicable to the traditional DP, and in such a case, is equivalent to the moments accountant (Abadi et al. 2016) for the subsampled Gaussian mechanism and Rényi accountant (Mironov 2017). Let us briefly outline the method.

Denote by s_t and s'_t signals sent by agents x and x' in round t correspondingly. A set of signals sent in rounds 1 through T is denoted by s_{1:T}. In the context of PALMA, these signals represent either an attempt to acquire a certain resource, or a back-off from a previously contested resource. Let q be the probability of any agent to ‘act’ in any given round, i.e., send a signal (sub sampling probability).

Using the derivations by Triastcyn and Faltings (2020), we can employ the following concentration inequality to obtain (ε, δ)-LDP bound for T rounds:

\[ \Pr[L ≥ ε] ≤ e^{AD_{1+1}(p(s_t|x)p(s'_t|x'))} - λε ≥ δ, \]  

where \( D_λ(\cdot∥\cdot) \) is the Rényi divergence, defined as

\[ D_λ(P∥Q) = \frac{1}{λ-1} \log \mathbb{E}_p \left[ \left( \frac{p(x)}{q(x)} \right)^λ \right] dx \]  

\[ = \frac{1}{λ-1} \log \mathbb{E}_q \left[ \left( \frac{p(x)}{q(x)} \right)^λ \right] dx, \]

and λ is a hyper-parameter (assume for simplicity λ ∈ N).

Following Triastcyn and Faltings (2020), we introduce the notion of privacy cost:

\[ c_t(λ) = λD_λ[p(s_t|x)p(s_t|x')), \]  

Given the subsampling probability q, the privacy cost of a single round \( t \) of PALMA for \( λ \in N \) can be computed as

\[ c_t(λ) = \max\{c^T_t(λ), c^R_t(λ)\}, \]

where

\[ c^T_t(λ) = \log \mathbb{E}_{k\sim B(λ+1, q)} \left[ \mathbb{E}_s \left[ \left( \frac{p(s_t|x')}{p(s_t|x)} \right)^λ \right] \right], \]

\[ c^R_t(λ) = \log \mathbb{E}_{k\sim B(λ+1, q)} \left[ \mathbb{E}_s \left[ \left( \frac{p(s_t|x)}{p(s_t|x')} \right)^λ \right] \right], \]

and \( B(λ, q) \) is the binomial distribution with \( λ \) experiments and probability of success \( q \) (Triastcyn 2020).

To bound the total privacy loss and compute ε from δ or vice versa, we can use an advanced composition theorem. As stated, the Bayesian accountant (Triastcyn and Faltings 2020), the moments accountant (Abadi et al. 2016) and the Rényi accountant (Mironov 2017) are equivalent in this case, resulting in:

\[ \log δ ≤ \sum_{t=1}^T c_t(λ) - λε, \]

\[ ε ≤ \frac{1}{λ} \sum_{t=1}^T c_t(λ) - \frac{1}{λ} \log δ. \]

5 Evaluation

In this section we evaluate PALMA in a realistic vehicle-passenger matching scenario in a ridesharing application. We focus on solution quality (social welfare, i.e., sum of utilities \( \sum_{u \in U} U_u(s) \)) and level of privacy (ε given δ = 1e−5).

Each problem instance is run 16 times. We report the average value for the social welfare, the average value for the median of ε, and the maximum value of ε out of all the runs. Error bars represent one standard deviation (SD) of uncertainty.

PALMA’s parameters were set to \( γ = ζ_B = ζ_S = 0.05, \) \( q = 0.1 \) (Eq. 3 4 1) and the subsampling probability, respectively, and each agent had a privacy budget of \( ε = 3 \).

The Rényi divergence’s hyper-parameter was set to \( λ = 32 \).

5.1 Motivation

The emergence and widespread use of ridesharing in recent years has had a profound impact on urban transportation. Normally the process is facilitated by a centralized operator,
which requires accurate location information of passengers and vehicles, which raises privacy concerns. Such a problem is ideal to showcase PALMA: (i) Ridesharing (and mobility-on-demand in general) occurs in unboundedly large settings. It is obvious that if a ridesharing provider attempts to naively add noise to protect all users (independently of their characteristics) with the same guarantee, the achieved social welfare will be as good as a random solution in large-scale environments. (ii) It is reasonable to assume an informed attacker (e.g., one that knows the residing city of an individual), and users may be willing to reveal approximate location information. Moreover, contrary to other approaches (e.g., (Prorok and Kumar 2017; Fioretto, Lee, and Van Hen- tenryck 2018)) PALMA is decentralized and employs local DP, providing privacy against a malicious data curator.

The Dynamic Ridesharing and Fleet Relocation problem can be decomposed into three maximum-weight matching sub-problems – request to request matching, shared ride to vehicle matching, and idle vehicle relocation – all of which can be solved efficiently by PALMA (Danassis et al. 2019). In this test-case we will focus on the second sub-problem; passenger to vehicle matching.

5.2 Setting

Our evaluation setting is specifically designed to resemble reality as closely as possible, following the modeling of (Danassis et al. 2019). We have used the yellow taxi trip records of 2016, provided by the NYC Taxi and Limousine Commission (TLC 2016). For every request, the dataset provides amongst others the geo-location coordinates. The set of agents $N$ is composed by the requests at the Manhattan area from 08:00:00 - 08:00:10, on January 15th (51 requests in total). The set of resources $R$ includes 51 vehicles scattered across the map. To avoid cold start, the position of each of the taxis was set to the drop-off geo-location of the last 51 requests prior to 08:00:00. We used the Manhattan distance as a distance function (using the Haversine formula) to calculate the distance in each coordinate, as it has been found to be a close approximation of the actual driving distance in Manhattan (Danassis et al. 2019). The utility function for each agent was given by Eq. (13) where $\alpha$ controls the steepness in utility loss as we increase the distance between an agent and a resource and $d(n,r)$ denotes the distance between agent $n$ and resource $r$. We set $\alpha = 4000$ (recall that distance is in m).

$$u_n(r) = e^{-\frac{d(n,r)}{\alpha}}$$ (13)

The area of operation of the ridesharing company is divided into fixed square regions of edge length $\ell$. The proposed Piecewise Local Differential Privacy demands that users belonging to the same region be indistinguishable from the attacker’s point of view. We have evaluated PALMA in test-cases with length $\ell \in \{1000, 2000, 3000, 4000\}$ m, which roughly correspond to an area of $\{45.6, 182.5, 410.5, 730\}$ city blocks. Figure 1 offers a visual representation of the setting.

Each agent $n$ independently calculates the locations of two potential agents $n'$, and $n''$ inside his region that result in the worst privacy loss given the agent’s selection and back-off distributions (Eq. 1 and 4, respectively). Using this information, each agent is able to keep track of his privacy budget at every time-step, and calculate his total $\varepsilon$ after convergence.

To compute the optimal – in terms of social welfare – solution, we used the non-private, centralized Hungarian algorithm (Kuhn 1955).

5.3 Simulation Results

Social Welfare  PALMA loses approximately $10.8\% \pm 2\%$ ($\ell = 1000$) to $26\% \pm 0.3\%$ ($\ell = 4000$) in social welfare compared to the non-private, optimal solution for increasing region edge length ($\ell$).

Figure 1: A visual representation of the regions ($\{D_i\}$) created by $\varphi(\cdot)$ of PLDP specifically for the ridesharing application. Red dots denote the edge points of each region ($\ell = 4000$). Orange dots represent the agents (requests), and blue dots represent the resources (vehicles).

Figure 2: Loss in social welfare compared to the non-private, optimal solution for increasing region edge length ($\ell$).

A ridesharing company can operate across multiple cities, counties, or even continents.

https://en.wikipedia.org/wiki/Haversine_formula

https://en.wikipedia.org/wiki/City_block
Figure 3: Maximum (orange line) and median (blue line) per-agent $\varepsilon$ for increasing region edge length ($\ell$). The shaded area represents the range between the maximum and minimum value of the median.

Figure 4: Histogram of per-agent $\varepsilon$ for privacy region edge length $\ell = 4000$. We include all 16 runs (16 runs $\times$ 51 agents = 816 data points in total).

Figure 5: Loss in social welfare for decreasing privacy budget ($\varepsilon$). The region edge length is set to $\ell = 1000$.

is because the majority of the agents converge fast (Danassis, Filos-Ratsikas, and Faltings 2019), thus only a small percentage of them will exhaust their privacy budget. In fact, more than half of the total agents (455 out of the 816, or 55.8%) have $\varepsilon \leq 0.5$. It is clear that the vast majority of agents benefit from really high degree of privacy. This is consistent with the findings of Triastcyn (2020) that privacy loss is high for only a small portion of data/agents.

**Privacy Budget vs. Social Welfare** Given that only a small number of agents deplete their privacy budget, we studied the effect of an even stricter privacy budget (i.e., higher level of privacy) to the solution quality. Figure 5 depicts the achieved social welfare for decreasing privacy budget ($\varepsilon$). Restricting the budget from $\varepsilon = 3$ to $\varepsilon = 1$ results in only $\approx 0.69\%$ additional loss in social welfare. Finally, even under really strict privacy requirements ($\varepsilon = 0.5$), PALMA achieves low loss in social welfare ($-19\%$).

**6 Conclusion**

As we bridge the gap between physical and cyber worlds, matching problems have become ubiquitous in everyday life, which brings about significant privacy risks and the potential to reveal highly sensitive information of users.

In this paper, we consider the problem of hiding the utility function in a maximum-weight matching problem, and motivate and develop an ‘application-aware’ privacy model, Piecewise Local Differential Privacy (PLDP), which takes into account the ‘distance’ between two utility functions. This ensures indistinguishability between agents with similar preferences, under an informed attacker.

We also propose PALMA, a privacy-preserving heuristic for maximum-weight matching in real-world, large-scale applications. PALMA is decentralized, requires no inter-agent communication, converges in constant time under reasonable assumptions, and provides a strong level of privacy ($\varepsilon \leq 3$ and a median value of 0.44 for $\delta = 1e^{-3}$ in a vehicle-passerenger matching test-case using real data).

PALMA is applicable in a wide-range of domains. Examples include mobility-on-demand systems (e.g., ridesharing), resource allocation in urban environments (e.g., matching charging stations to autonomous vehicles), intelligent infrastructure & IoT devices (e.g., channel allocation in wireless networks), paper assignment in conference management systems (reviewers to manuscripts), meeting scheduling, and so on.
References
Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H. B.; Mironov, I.; Talwar, K.; and Zhang, L. 2016. Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 308–318.

Andrés, M. E.; Bordenabe, N. E.; Chatzikokolakis, K.; and Palamidessi, C. 2013. Geo-Indistinguishability: Differential Privacy for Location-Based Systems. In Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, CCS ’13, 901–914. New York, NY, USA: Association for Computing Machinery. ISBN 9781450324779. doi:10.1145/2508859.2516735.

Balle, B.; Barthe, G.; and Gaboardi, M. 2018. Privacy amplification by subsampling: Tight analyses via couplings and divergences. In Advances in Neural Information Processing Systems, 6277–6287.

Balle, B.; Kairouz, P.; McMahan, H. B.; Thakkar, O.; and Thakurta, A. 2020. Privacy amplification via random check-ins. arXiv preprint arXiv:2007.06605.

Chatzikokolakis, K.; Andrés, M. E.; Bordenabe, N. E.; and Palamidessi, C. 2013. Broadening the Scope of Differential Privacy Using Metrics. In De Cristofaro, E.; and Wright, M., eds., Privacy Enhancing Technologies, 82–102. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-642-39077-7.

Danassis, P.; Filos-Ratsikas, A.; and Faltings, B. 2019. Anytime Heuristic for Weighted Matching Through Altruism-Inspired Behavior. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, 215–222. doi:10.24963/ijcai.2019/31. URL https://doi.org/10.24963/ijcai.2019/31

Danassis, P.; Sakota, M.; Filos-Ratsikas, A.; and Faltings, B. 2019. Putting Ridesharing to the Test: Efficient and Scalable Solutions and the Power of Dynamic Vehicle Relocation. ArXiv: 1912.08066.

Dwork, C. 2006. Differential Privacy. In Bugliesi, M.; Preneel, B.; Sassone, V.; and Wegener, I., eds., Automata, Languages and Programming, 1–12. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-540-35908-1.

Dwork, C.; Kenthapadi, K.; McSherry, F.; Mironov, I.; and Naor, M. 2006a. Our data, ourselves: Privacy via distributed noise generation. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, 486–503. Springer.

Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. 2006b. Calibrating noise to sensitivity in private data analysis. In Theory of cryptography conference, 265–284. Springer.

Dwork, C.; Roth, A.; et al. 2014. The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science 9(3-4): 211–407.

Edmonds, J. 1965. Maximum matching and a polyhedron with 0, 1-vertices. Journal of research of the National Bureau of Standards B.

Fioretto, F.; Lee, C.; and Van Hentenryck, P. 2018. Constrained-Based Differential Privacy for Mobility Services. In Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS ’18, 1405–1413. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Giordani, S.; Lujak, M.; and Martinelli, F. 2010. A distributed algorithm for the multi-robot task allocation problem. In International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Springer.

Ismail, S.; and Sun, L. 2017. Decentralized hungarian-based approach for fast and scalable task allocation. In 2017 Int. Conf. on Unmanned Aircraft Systems (ICUAS).

Kuhn, H. W. 1955. The Hungarian method for the assignment problem. Naval Research Logistics.

Lovász, L.; and Plummer, M. D. 2009. Matching theory, volume 367. American Mathematical Soc.

Mironov, I. 2017. Rényi differential privacy. In 2017 IEEE 30th Computer Security Foundations Symposium (CSF), 263–275. IEEE.

Prorok, A.; and Kumar, V. 2017. Privacy-preserving vehicle assignment for mobility-on-demand systems. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1869–1876. IEEE.

Roth, A. E.; Sönmez, T.; and Unver, M. U. 2005. Pairwise kidney exchange. Journal of Economic theory 125(2): 151–188.

Stone, P.; Kaminka, G. A.; Kraus, S.; and Rosenschein, J. S. 2010. Ad Hoc Autonomous Agent Teams: Collaboration without Pre-Coordination. In Proceedings of the Twenty-Fourth Conference on Artificial Intelligence.

Su, H.-H. 2015. Algorithms for Fundamental Problems in Computer Networks.

TLC. 2016. NYC Taxi and Limousine Commission Trip Record Data. https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page Accessed: 2019-11-10.

Triastcyn, A. 2020. Data-Aware Privacy-Preserving Machine Learning. Ph.D. thesis, Lausanne. doi:10.5075/epfl-thesis-7216. URL http://infoscience.epfl.ch/record/280791

Triastcyn, A.; and Faltings, B. 2019. Federated Learning with Bayesian Differential Privacy. In 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, December 9-12, 2019, 2587–2596. IEEE. doi: 10.1109/BigData47090.2019.9005465. URL https://doi.org/10.1109/BigData47090.2019.9005465

Triastcyn, A.; and Faltings, B. 2020. Bayesian Differential Privacy for Machine Learning. In 37th International Conference on Machine Learning.

Warner, S. L. 1965. Randomized response: A survey technique for eliminating evasive answer bias. Journal of the American Statistical Association 60(309): 63–69.

Zavlanos, M. M.; Spesivtsev, L.; and Pappas, G. J. 2008. A distributed auction algorithm for the assignment problem. In Decision and Control, 2008. IEEE.