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 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 already know coarser-grained information (e.g., city of residence in a mobility-on-demand system, or reviewer expertise in a paper assignment problem) because it is likely pubic or easily available, and thus, does not need to be hidden, 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 indistinguishably between reviewers on different fields).

We introduce *Piecewise Local Differential Privacy* (PLDP), a privacy model designed to protect the utility function. PLDP allows for 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 mobility-on-demand and a paper assignment scenario, using *real data* in both, and demonstrate that it provides *strong privacy*, $\varepsilon \leq 3$ and median $\varepsilon = 0.4$ across agents, and *high quality* matchings (up to 85% of 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 Ünver 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. (P)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

2020; Triastcyn 2020) and distance-based generalisations of DP (Chatzikokolakis et al. 2013; Andrés et al. 2013). Despite notable similarities with the latter, there is an important difference. Instead of being centered around the agent, privacy-protected regions are predefined by some function $\varphi(\cdot)$ that partitions the space of possible outcomes $\mathcal{D}$ into subspaces $\{\mathcal{D}_i\}$. Importantly, 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 ε over the iterations. We provide more details in Section 2. Finally, we combine PLDP with ALMA to create a decentralized, privacy-preserving heuristic (PALMA) for large-scale maximum-weight matching problems.

1.1 Our Contributions

(1) We introduce *Piecewise Local Differential Privacy* (PLDP), a variant of differential privacy designed to protect the utility function in multi-agent applications. PLDP enables significant improvements in solutions quality and strong theoretical privacy guarantees, while being applicable in *real-world, unboundedly large settings*.

(2) We propose *PALMA*, a privacy-preserving heuristic for *large-scale* maximum-weight matching applications.

(3) We evaluate PALMA in a mobility-on-demand and a paper assignment scenario, using *real data*. PALMA is able to provide a high degree of privacy, $\varepsilon \leq 3$ and a median value of 0.4 across agents for $\delta = 1e^{-5}$, and matchings of high quality (up to 85% of the non-private optimal).

As a bonus contribution, we extend the existing tight adaptive accounting for Gaussian subsampled mechanisms to general subsampled mechanisms.

1.2 Related Work

Finding a maximum-weight matching is one of the best-studied combinatorial optimization problems (Su 2015; Lovász and Plummer 2009). There is a plethora of centralized polynomial time algorithms (e.g., Hungarian (Kuhn 1955), blossom (Edmonds 1965)). 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., (Ismail and Sun 2017; Zavlanos, Spesivtsev, and Pappas 2008)) 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 utilities, plans, and preferences creates high overhead. ALMA on the other hand achieves *constant* in the total problem size running time – under reasonable assumptions – 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[1]. 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. In fact, it is impossible to have both meaningful social welfare and privacy guarantees in matching problems under (L)DP (Hsu et al. 2014). (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* which 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, *Piecewise Local Differential Privacy (PLDP)*.

2.1 Definition

We consider a randomized function $\mathcal{M} : \mathcal{D} \to \mathcal{A}$ with domain $\mathcal{D}$ and range $\mathcal{A}$. In the context of matching problems

[1] For a more comprehensive overview of DP and DP mechanisms, we refer the reader to (Triastcyn 2020; Dwork, Roth et al. 2014).

in multi-agent systems, $\mathcal{D}$ is the space of all utility functions and $\mathcal{A}$ is the action space.

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

$$\Pr\left[\mathcal{M}(x) \in \mathcal{S} \mid x \in \mathcal{D}_i\right] \leq e^{\varepsilon} \Pr\left[\mathcal{M}(x') \in \mathcal{S} \mid x' \in \mathcal{D}_i\right] + \delta.$$

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 φ such that the initial location space is mapped into ZIP codes. Then, $(\varepsilon, \delta, \varphi)$-PLDP guarantee would certify that the reported statistic is (ε, δ)-locally private within every ZIP code. However, it would not tell us anything about privacy of the reported statistic outside the given ZIP code.

2.2 Privacy Properties

Note that PLDP is a straightforward relaxation of local privacy and all the properties of LDP are satisfied within subdomains $\mathcal{D}_i$. In order to see that this is true, it is sufficient to consider the following. Once the space $\mathcal{D}$ has been partitioned, the PLDP definition is equivalent to the LDP definition within each sub-space $\mathcal{D}_i$. Hence, basic properties of (L)DP, such as *composition*, *post-processing*, and *group privacy*, as well as several instances of *advanced composition* (Dwork, Roth et al. 2014; Abadi et al. 2016), will also hold for any pair x, x' from a given $\mathcal{D}_i$, as long as these points do not dynamically change sub-spaces between applications of the privacy mechanism. The latter condition *is satisfied in all considered scenarios*: every new matching routine starts with a fresh set of agents with random identifiers, and agents do not change their utilities during the matching process.

2.3 Advantages of PLDP

PLDP closely resembles another well-known privacy notion, geo-indistinguishability (Andrés et al. 2013), which is based on a generalization of DP (Chatzikokolakis et al. 2013). Nonetheless, there is a notable distinction. To put it in terms of the definition above, in geo-indistinguishability, the region within which privacy is protected is centered at x. In our definition, these regions are predefined by φ. As a downside, our privacy guarantee is limited to the given region rather than fading gradually with increasing region radius. However, this subtle difference becomes crucial in real-world applications due to composition properties. To the best of our knowledge, in spite of conveniently adopting the use of distances between inputs to adjust levels of privacy guarantees, geo-indistinguishability has only been proven to satisfy basic composition. As a result, ε grows linearly with the number of privacy mechanism invocations. It is not sufficiently tight for iterative AI and ML applications, which typically require a lot of repetitive applications of privacy mechanisms (Abadi et al. 2016). On the other hand, PLDP allows to use *tighter composition theorems* developed for the conventional DP, *reducing the growth* of ε from linear w.r.t. the total number of algorithm iterations T to $\mathcal{O}(\sqrt{T})$ (Abadi et al. 2016).

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

In this section we introduce *PALMA* (Privacy-preserving ALtruistic MAtching), a privacy-preserving adaptation of ALMA (Danassis, Filos-Ratsikas, and Faltings 2019). We start by describing the problem of finding a maximum-weight matching[2]. Finally, we describe the employed privacy mechanisms, and the privacy accounting method.

3.1 The Assignment Problem

The assignment problem refers to finding a maximum-weight matching in a weighted bipartite graph, $\mathcal{G} = \{\mathcal{N} \cup \mathcal{R}, \mathcal{E}\}$. In the studied scenario, $\mathcal{N} = \{1, \dots, N\}$ agents compete to acquire $\mathcal{R} = \{1, \dots, R\}$ 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 sum of utilities.

3.2 Learning Rule

We assume that each agent is interested in (potentially) a subset of the total resources $\mathcal{Q}^n \subseteq \mathcal{R}$. Let $\mathcal{A} = \{Y, A_{r_1}, \dots, A_{r_{Q^n}}\}$ 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. PALMA is run *independently and in parallel by all the agents*. An agent running PALMA converges to a resource through repeated trials, specifically:

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[3], the colliding parties back-off with some probability, $P_B^n(\cdot)$. Otherwise, if $g = Y$, the agent chooses a resource r for monitoring according to probability , $P_S^n(\cdot)$. If the resource is free, he sets $g \leftarrow A_r$. The pseudo-code of PALMA can be found in Algorithm 1.

Resource Selection Distribution In the original implementation of ALMA, each agent sorts the resources in decreasing order of utility $(r_1, \dots, r_i, \dots, r_R)$. Then, he 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

[2] For simplicity, we focus on bipartite graphs, but (P)ALMA can be applied in general graphs as well (see (Danassis et al. 2019)).

[3] We assume that agents can observe feedback from their environment to inform collisions and detect free resources (e.g., by the use of *sensors*, or by a *single bit* feedback from the resource).

highest social welfare, 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 at random fashion, where resource r_i is selected with probability $\frac{u_n(r_i)}{\sum_{r\in\mathcal{R}} u_n(r)}$. 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 ($|\mathcal{R}| \to \infty$) where agents are interested only in resources that are physically close to them, the majority of resources would have utility ≈ 0. If we select a resource in a weighted 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 aforedescribed two approaches. Let $\mathcal{N}^n$ denote the set of every possible agent that belongs to the same region of utility space as n, i.e., $\mathcal{N}^n = \{n' : u_{n'}(\cdot) \in \mathcal{D}_i \wedge u_n(\cdot) \in \mathcal{D}_j \Rightarrow i = j\}$. We refer to $\mathcal{N}^n$ as the set of neighbors of n. Note that the neighbors of an agent do not need to be in $\mathcal{N}$, we account for every potential agent (i.e., $\cup_{n\in\mathcal{N}}\mathcal{N}^n \supset \mathcal{N}$). The *neighbors* are the *set of agents that PLDP guarantees indistinguishability*. Then, agent n generates the sets $(\mathcal{R}_1^n, \dots, \mathcal{R}_i^n, \dots, \mathcal{R}_R^n)$, where the set $\mathcal{R}_i^n$ contains the i^{th} most preferred resource of each neighbor, i.e., $\mathcal{R}_i^n = \cup_{\forall n'\in\mathcal{N}^n}\{r_i^{n'}\}$.

Agent n moves in a *sequential* manner from set to set (starting from the set of the most preferred resources, $\mathcal{R}_1^n$, and looping back to it after $\mathcal{R}_R^n$). The resource selection is performed in a *weighted at random* fashion in the sets $\mathcal{R}_i^n$. Specifically, at step $s = t \mod R$, where t is the current time-step, agent n will select resource $r_i \in \mathcal{R}_s^n$ with probability given by (line 15 of Algorithm 1):

$$P_S^n(i, s, \zeta_S) = (1-\zeta_S)\frac{u_n(r_i)}{\sum_{r\in\mathcal{R}_s^n} u_n(r)} + \frac{\zeta_S}{|\mathcal{R}_s^n|} \tag{1}$$

where ζ_S tunes the magnitude of the introduced randomness.

Back-off Distribution The back-off probability, $P_B^n(\cdot) : \mathcal{R} \to [0,1]$ (line 8 of Algorithm 1), is computed individually and locally based on each agent's expected utility loss that he will incur if he switches:

$$loss_n(i, s) = u_n(r_i) - \sum_{r_j\in\mathcal{R}_{s+1}^n} \frac{u_n(r_j)}{\sum_{r\in\mathcal{R}_{s+1}^n} u_n(r)} u_n(r_j) \tag{2}$$

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

$$f(loss) = \begin{cases} 1-\gamma, & \text{if } loss \leq \gamma \\ \gamma, & \text{if } 1 - loss \leq \gamma \\ 1 - loss, & \text{otherwise} \end{cases} \tag{3}$$

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

$$P_B^n(i, \zeta_B) = (1-\zeta_B) f(loss_n(\cdot)) + \frac{\zeta_B}{2} \tag{4}$$

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 $P_S^n(\cdot)$, and examine its availability (line 15 of Algorithm 1).

Algorithm 1 PALMA:
Privacy-preserving ALtruistic MAtching heuristic.

s	Current step (indicates a specific set $\mathcal{R}_s^n$)
g	Specifies which resource to access
$\{Y, A_{r_1}, \dots, A_{r_R}\}$	Y refers to yielding, and A_r refers to accessing resource r
$P_S^n(\cdot) : \mathcal{R} \to [0,1]$	Resource selection probability distribution
$P_B^n(\cdot) : \mathcal{R} \to [0,1]$	Back-off probability distribution
c	Accumulated privacy cost
c_{max}	Highest possible privacy cost for selection or back-off
B_n	Privacy budget

```
 1: Initialize $s \leftarrow 1$, $g \sim P_S^n(\cdot)$, $c \leftarrow 0$, $c_{max}$, Accountant($q, \lambda$)
 2: procedure PALMA(subsampled)
 3:     if subsampled then
 4:         if $g = A_r$ then
 5:             Agent $n$ attempts to acquire $r$
 6:             if Collision($r$) then
 7:                 if Accumulate($c, c_{max}, q, \lambda$) $< B_n$ then
 8:                     Back-off (set $g \leftarrow Y$) with probability $P_B^n(\cdot)$
 9:                     $c \leftarrow$ Accumulate($c, c_{max}, q, \lambda$)
10:                 else
11:                     Back-off (set $g \leftarrow Y$) with probability $0.5$
12:         else ($g = Y$)
13:             $s \leftarrow (s + 1) \mod R$
14:             if Accumulate($c, c_{max}, q, \lambda$) $< B_n$ then
15:                 Agent $n$ monitors $r \sim P_S^n(\cdot)$
16:                 $c \leftarrow$ Accumulate($c, c_{max}, q, \lambda$)
17:             else
18:                 Agent $n$ monitors $r \sim \mathcal{U}(\mathcal{R}_s^n)$
19:             if Free($r$) then set $g \leftarrow A_r$
20:     return $r$, such that $g = A_r$
21: Output $r$, such that $g = A_r$, and $(\varepsilon, \delta) \leftarrow$ getPrivacy($c$)
```

Bounding the Set of Desirable Resources In real-world applications there is typically a cost associated with acquiring a resource Thus, each agent is typically interested in a subset of the total resources, i.e., $\mathcal{Q}^n \subset \mathcal{R}$. This results in faster convergence (*constant* time, see Section B), but can also potentially lead to higher social welfare[4]. The sets $(\mathcal{R}_1^n, \dots, \mathcal{R}_i^n, \dots, \mathcal{R}_R^n)$ can be contracted in similar manner.

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.

- **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 ζ_S and ζ_B, respectively (see Eq. 1 and 4). The idea is that the agent first flips a coin to decide whether to act truthfully. Then, with probability $1 - \zeta_S$ (or $1 - \zeta_B$), the agent plays according to its true selection (or back-off) function;

[4] The agent will loop back to $\mathcal{R}_1^n$, increases his chances of winning a high utility resources, instead of moving through a large number of undesirable resources.

with probability ζ_S (or ζ_B), the agent plays uniformly at random.

- **Amplification by Sampling** We amplify privacy through sampling (Abadi et al. 2016; Balle, Barthe, and Gaboardi 2018). At each time-step, we sample a subset of agents to play (step 3 of Algorithm 1), according to subsampling probability q. Currently the sampling is performed centrally. We can employ a decentralized random subsampling (similar to (Balle et al. 2020)) as well, but this is out of the scope of this work.

Lastly, each agent has a privacy budget of $\varepsilon = B_n$. Upon depletion in the course of using the above mechanisms, the agent will play *random* actions (see lines 7 & 14 of Alg. 1).

Note also that each agent can select the fragmentation function $\varphi(\cdot)$ of PLDP and adjust the size of the neighborhood $\mathcal{N}^n$ according to his privacy needs.

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 (e.g., (Dwork, Roth et al. 2014)). We employ the accounting framework introduced by Triastcyn and Faltings (2020) and extend it to generic subsampled mechanisms. While developed for the notion of Bayesian DP, this framework is applicable to the traditional DP as well, 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.

Let σ_t and σ'_t denote *signals* sent by agents x and x' in time-step t, and ξ_t any auxiliary information. A set of signals (auxiliary information) sent in time-steps 1 through T is denoted by $\sigma_{1:T}$ ($\xi_{1:T}$). In the context of PALMA, these signals represent either an attempt to acquire a resource, or a back-off from a previously contested resource, while the auxiliary information corresponds to s (which determines the set of resources $\mathcal{R}_s$, see Eq. 1, 4). Let q be the probability of an agent to 'act' in any given time-step (subsampling 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:

$$\Pr[L \geq \varepsilon] \leq e^{\lambda \mathcal{D}_{\lambda+1}[p(\sigma_{1:T}|\xi_{1:T},x) \| p(\sigma_{1:T}|\xi_{1:T},x')] - \lambda\varepsilon} = \delta,$$

where L denotes the privacy loss random variable and $\mathcal{D}_\lambda(\cdot\|\cdot)$ is the Rényi divergence of order λ (see Section C).

We also introduce the notion of *privacy cost*:

$$c_t(\sigma_t, \xi_t, x, x', \lambda) \triangleq \lambda \mathcal{D}_{\lambda+1}[p(\sigma_t|\xi_t, x) \| p(\sigma_t|\xi_t, x')].$$

Finally, we can formulate the theorem for accounting general subsampled mechanisms.

Theorem 1. *Given the subsampling probability q, the privacy cost of one application of privacy mechanism (a single round t of PALMA) for $\lambda \in \mathbb{N}$ can be computed as*

$$c_t(\sigma_t, \xi_t, x, x', \lambda) = \max\{c_t^L(\sigma_t, \xi_t, x, x', \lambda), c_t^R(\sigma_t, \xi_t, x, x', \lambda)\},$$

where

$$c_t^L(\sigma_t, \xi_t, x, x', \lambda) = \log \mathbb{E}_{k \sim B(\lambda+1,q)} \left[\mathbb{E}_{s_t} \left[\left(\frac{p(\sigma_t|\xi_t, x')}{p(\sigma_t|\xi_t, x)} \right)^k \right] \right],$$

$$c_t^R(\sigma_t, \xi_t, x, x', \lambda) = \log \mathbb{E}_{k \sim B(\lambda+1,q)} \left[\mathbb{E}_{s_t} \left[\left(\frac{p(\sigma_t|\xi_t, x)}{p(\sigma_t|\xi_t, x')} \right)^k \right] \right],$$

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

Proof. See Section D. □

4.1 PALMA's Privacy Cost

Every matching game starts with a fresh set of agents with random identifiers. Each agent computes (*once*, and *off-line*) the highest possible privacy cost at any round (c_{max}), i.e., the maximum value between the worst possible privacy cost during resource selection and back-off:

$$c_{max} = \max \begin{cases} \max\limits_{\xi_t \in \{1,\dots,R\}} \max\limits_{x' \in \mathcal{N}^x} \max\limits_{\sigma_t \in \mathcal{R}^x_{\xi_t} \sim P^n_S(\cdot)} c_t(\cdot) \\ \max\limits_{\xi_t \in \{1,\dots,R\}} \max\limits_{x' \in \mathcal{N}^x} \max\limits_{\sigma_t \in \mathcal{R}^x_{\xi_t} \sim P^n_B(\cdot)} c_t(\cdot) \end{cases} \tag{5}$$

The agents do not change their utilities during the matching process (i.e., the distributions $P^n_S(\cdot)$ and $P^n_B(\cdot)$ stay fixed), thus each agent can *compute a priori* the total privacy cost (*worst case privacy guarantees*) and the maximum number of rounds until the budget B_n is exhausted and he has to play randomly. Agents can then adjust their privacy parameters accordingly. The actual privacy loss is accounted on the fly during execution (see lines 9 and 16 of Alg. 1).

To bound the total privacy loss over multiple rounds and compute ε from δ or vice versa, we can use an advanced composition theorem. As stated, the advanced compositions theorem for the 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:

$$\log \delta \leq \sum_{t=1}^{T} c_{max}(\cdot) - \lambda\varepsilon \qquad \varepsilon \leq \frac{1}{\lambda} \sum_{t=1}^{T} c_{max}(\cdot) - \frac{1}{\lambda} \log \delta$$

5 Evaluation

We evaluate PALMA in a *mobility-on-demand* and a *paper assignment* application, using *real-data* for both. We focus on the social welfare (sum of utilities $\sum_{n \in \mathcal{N}} u_n(\cdot)$) and level of privacy ($\varepsilon$ given $\delta = 1e^{-5}$). 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. Error bars represent one standard deviation. We set $\gamma = \zeta_B = \zeta_S = 0.05$, $q = 0.1$, $\lambda = 32$.

5.1 Test-Case 1: Mobility on Demand

Motivation The emergence and widespread use of mobility-on-demand (MOD) services (e.g., *ridesharing* platforms like Uber or Lyft) in recent years has had a profound impact on urban transportation. Normally the process is facilitated by a centralized operator, that requires accurate location information of passengers and vehicles, which

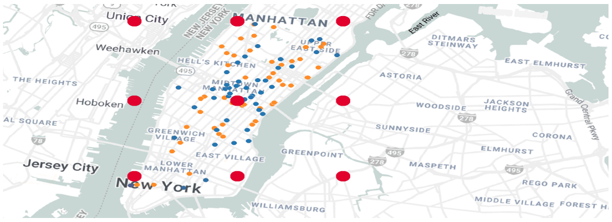

Figure 1: A visual representation of the regions ($\{\mathcal{D}_i\}$) of PLDP for the mobility-on-demand application. Red dots denote the edge points of each region ($\ell = 4000$). Orange dots represent the agents (requests), and blue dots represent the resources (vehicles).

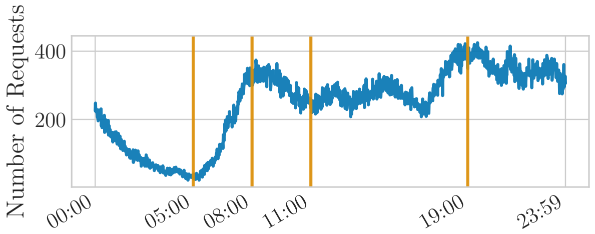

Figure 2: Request per minute in Manhattan on Jan 15, 2016.

raises privacy concerns. Such a problem is ideal to showcase PALMA: (i) Mobility-on-demand applications occur in unboundedly large settings. It is obvious that if a mobility-on-demand 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[5]. (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 Hentenryck 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, using PLDP and PALMA to provide a *scalable*, *on-device*, distributed solution that protects user preferences (i.e., *user location* in this context).

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.

We report results on four 30s instances on a typical day (January 15th). These instances were selected to represent various distributions of demand (see Figure 2). 05:00:00 - 05:00:30 represents the lowest demand, 08:00:00 - 08:00:30 and 19:00:00 - 19:00:30 represent the two rush hours (in the morning and evening, respectively), and finally, 11:00:00 - 11:00:30 represents a mid-day low. We selected 30s periods, because in practice the granularity of in-batches approaches for on-demand mobility services is between 10s[6] to 30s (Alonso-Mora et al. 2017; Riley, van Hentenryck, and Yuan 2020). It is important to stress this *does not affect the scalability* of the proposed approach. Running PALMA for a day, for example, would result in running 24 × 60 × 2 batches (as was done in (Danassis et al. 2019)). Assuming similar distributions for requests and vehicles[7], the *social welfare* and *privacy cost* of each agent will remain the *same*, since the privacy cost (Eq. 5) *only depends on the size of the region* $\mathcal{D}_i$.

The set of agents $\mathcal{N}$ is composed by the requests in the Manhattan area (17, 154, 116, and 174 requests in total on each of the evaluated batches). The set of resources $\mathcal{R}$ includes an equal number of vehicles scattered across the map. To avoid cold start, the position of each of the vehicles was set to the drop-off geo-location of the last (prior to the start time of the simulation) x requests (where x is the number of vehicles in each case). 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 is $u_n(r) = e^{-\frac{d(n,r)}{\alpha}}$, where $\alpha = 4000$ controls the steepness and $d(n, r)$ denotes the distance between agent n and resource r (in m).

The area of operation of the ridesharing company is divided into fixed square regions of edge length ℓ (which correspond to the $\mathcal{D}_i$). 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.

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

5.2 Simulation Results

Social Welfare PALMA loses between $14.6 \pm 3\%$ ($\ell = 1000$) to $30.1 \pm 4\%$ ($\ell = 4000$) in social welfare compared to the non-private, optimal solution (Figure 3). As a baseline, we have plotted the maximally private solution (i.e., the centralized random), which losses $49.4 \pm 2\%$.

[5] A mobility-on-demand company can operate across multiple cities, counties, or even continents.

[6] We also ran the same instances in batches of 10s obtaining *better results* (in social welfare), but opted to present the worst case.

[7] A reasonable assumption given that our choice of evaluated distributions cover all the extremes, and a typical mid-day demand.

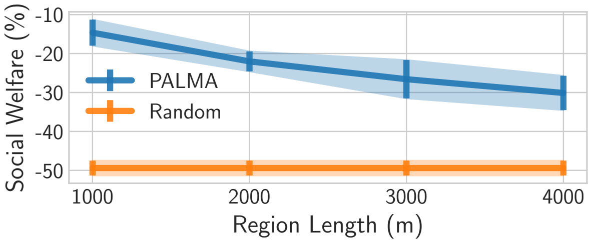

Figure 3: Loss in social welfare compared to the non-private, optimal solution for increasing region edge length (ℓ).

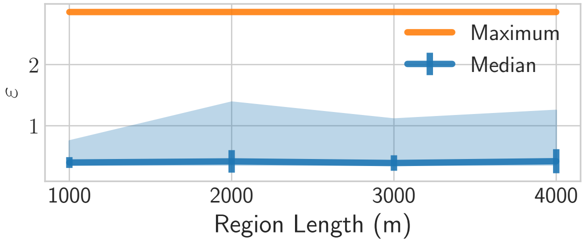

Figure 4: Maximum (orange line) and median (blue line) per-agent ε for increasing region edge length (ℓ). The shaded area represents the range between the maximum and minimum value of the median.

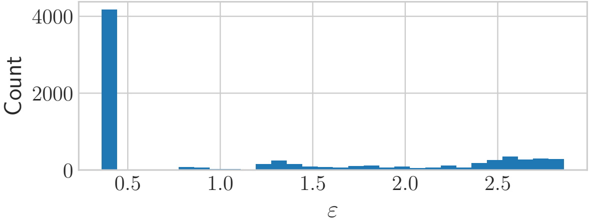

Figure 5: Histogram of per-agent ε for privacy region edge length $\ell = 4000$. We include all 16 runs (16 (runs) × (17 + 154 + 116 + 174) (agents) = 7376 data points in total).

In this test-case, every agent is interested in all the resource, i.e., $\mathcal{Q}^n = \mathcal{R}, \forall n \in \mathcal{N}$, but in a real-world scenario, there would be a bound to the size of the set of resources each agent is interested in (see Section B). The benefit would be twofold: (i) PALMA will converge in *constant* time, and (ii) if agents are only interested in resources in their vicinity, the social welfare will remain approximately *constant*, even if we run PALMA in *unboundedly large settings* (e.g., an entire country), assuming a fixed privacy region length (ℓ).

Privacy Figure 4 depicts the maximum (out of all the 16 runs) and median (average median value over the 16 runs) per-agent ε for increasing values of privacy region length ℓ. PALMA is able to achieve a *strong level of privacy* even in large-scale simulations. The average value of the median is 0.4. The maximum per-agent ε is bounded by the privacy budget (i.e., $\varepsilon = 3$). Recall that $\ell = 1000$ m corresponds to an area of 45.6 city blocks, and $\ell = 4000$ m is larger than the width of Manhattan (which is 3700 m wide at its widest).

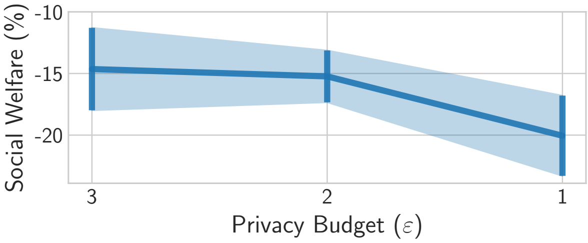

Figure 6: Loss in social welfare for decreasing privacy budget (ε). The region edge length is set to $\ell = 1000$.

Figure 5 plots the histogram of the per-agent ε for $\ell = 4000$. Only 3042 out of the 7376 agents (41.2%) have $\varepsilon > 1$, and only 1949 out of the 7376 agents (26.4%) have $\varepsilon > 2$. This is because the majority of the agents converge fast (Danassis, Filos-Ratsikas, and Faltings 2019), thus only a small percentage of them exhausts their budget. In fact, more than half of the total agents (4170 out of the 7376, or 56.5%) 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 budget (i.e., higher level of privacy) to the solution quality. Figure 6 depicts the achieved social welfare for decreasing privacy budget (ε). Restricting the budget from $\varepsilon = 3$ to $\varepsilon = 2$ results in only $\approx 0.6\%$ additional loss in social welfare, and from $\varepsilon = 3$ to $\varepsilon = 1$ in only $\approx 5.4\%$ additional loss.

6 Test-Case 2: Paper Assignment

We ran a second test-case, where we use PALMA to protect the reviewers' preferences during the paper assignment phase of a conference, using real data form (Karimzadehgan, Zhai, and Belford 2008). PALMA achieved similar results (loss in social welfare 26.4% (while the maximally private solution loses 71.5%); $\varepsilon \leq 3$ and a median value of 0.36). We refer to Section E for details.

7 Conclusion

Bridging the gap between physical and cyber worlds will bring 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 multi-agent coordination problems, 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. 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 median $= 0.4$).

A Appendix

A.1 Contents

In this appendix we include several details that have been omitted from the main text due to space limitations. In particular:

- In Section B we describe the computational complexity of PALMA.
- In Section C we provide the definition for the Rényi divergence.
- In Section D we prove Theorem 1 of the main text.
- In Section E, we provide a thorough account of the paper assignment test-case.
- Finally, in Section F we provide a short note on the societal impact of this work.

For narrative purposes, parts of the text of the main paper are repeated.

B Computational Complexity of PALMA

B.1 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., $\mathcal{Q}^n \subset \mathcal{R}$. 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 (*constant* time, see Section B.2), but can also potentially lead to higher social welfare[8]. The sets $(\mathcal{R}_1^n, \mathcal{R}_2^n, \dots, \mathcal{R}_x^n, \dots, \mathcal{R}_R^n)$ can be contracted in the same manner as before.

B.2 Convergence Speed

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., $\mathcal{Q}^n \subset \mathcal{R}$) and thus at each resource there is a bounded number of competing agents ($\mathcal{V}^r \subset \mathcal{N}$) 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 $|\mathcal{Q}^n|$, $|\mathcal{V}^r|$ to be constant functions of N, R), the convergence time is *constant* in the total problem size N, R.

The initialization of PALMA is linear to the size of the region, $\mathcal{O}(\max_i |\mathcal{D}_i|)$, but this can be done once off-line. Finally, the accounting of the privacy loss is $\mathcal{O}(1)$.

C Rényi Divergence Definition

The Rényi divergence of order λ is defined as (Triastcyn 2020):

$$\mathcal{D}_\lambda(P\|Q) = \frac{1}{\lambda-1}\log \mathbb{E}_p\left[\left(\frac{p(x)}{q(x)}\right)^{\lambda-1}\right]dx \tag{6}$$

$$= \frac{1}{\lambda-1}\log \mathbb{E}_q\left[\left(\frac{p(x)}{q(x)}\right)^{\lambda}\right]dx, \tag{7}$$

where λ is a hyper-parameter (assume for simplicity $\lambda \in \mathbb{N}$).

Analytic expressions for Rényi divergence exist for many common distributions and can be found in (Gil, Alajaji, and Linder 2013). (Van Erven and Harremos 2014) provides a good survey of Rényi divergence properties in general.

D Proof of Theorem 1

Let us first restate the theorem:

Theorem 1. *Given the subsampling probability q, the privacy cost of one application of privacy mechanism (a single round t of PALMA) for $\lambda \in \mathbb{N}$ can be computed as*

$$c_t(\sigma_t, \xi_t, x, x', \lambda) = \max\{c_t^L(\sigma_t, \xi_t, x, x', \lambda), c_t^R(\sigma_t, \xi_t, x, x', \lambda)\},$$

where

$$c_t^L(\sigma_t, \xi_t, x, x', \lambda) = \log \mathbb{E}_{k\sim B(\lambda+1,q)}\left[\mathbb{E}_{s_t}\left[\left(\frac{p(\sigma_t|\xi_t, x')}{p(\sigma_t|\xi_t, x)}\right)^k\right]\right],$$

$$c_t^R(\sigma_t, \xi_t, x, x', \lambda) = \log \mathbb{E}_{k\sim B(\lambda+1,q)}\left[\mathbb{E}_{s_t}\left[\left(\frac{p(\sigma_t|\xi_t, x)}{p(\sigma_t|\xi_t, x')}\right)^k\right]\right],$$

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

Proof. Similarly to the proof for Gaussian mechanism in (Triastcyn and Faltings 2020), let us consider the following expectation

$$\mathbb{E}_{s_t}\left[\left(\frac{p(s_t|\xi_t, \mathcal{N})}{p(s_t|\xi_t, \mathcal{N}')}\right)^{\lambda+1}\right] \tag{8}$$

$$= \mathbb{E}_{s_t}\left[\left(\frac{(1-q)p(s_t|\xi_t, x) + qp(s_t|\xi_t, x')}{p(s_t|\xi_t, x)}\right)^{\lambda+1}\right] \tag{9}$$

$$= \mathbb{E}_{s_t}\left[\left((1-q) + q\frac{p(s_t|\xi_t, x')}{p(s_t|\xi_t, x)}\right)^{\lambda+1}\right] \tag{10}$$

$$= \mathbb{E}_{s_t}\left[\sum_{k=0}^{\lambda+1}\binom{\lambda+1}{k}q^k(1-q)^{\lambda+1-k}\left(\frac{p(s_t|\xi_t, x')}{p(s_t|\xi_t, x)}\right)^k\right] \tag{11}$$

$$= \sum_{k=0}^{\lambda+1}\binom{\lambda+1}{k}q^k(1-q)^{\lambda+1-k}\mathbb{E}_{s_t}\left[\left(\frac{p(s_t|\xi_t, x')}{p(s_t|\xi_t, x)}\right)^k\right] \tag{12}$$

$$= \mathbb{E}_{k\sim\mathcal{B}(\lambda+1,q)}\left[\mathbb{E}_{s_t}\left[\left(\frac{p(s_t|\xi_t, x')}{p(s_t|\xi_t, x)}\right)^k\right]\right] \tag{13}$$

$$= \mathbb{E}_{k\sim\mathcal{B}(\lambda+1,q)}\left[e^{k\mathcal{D}_{k+1}(p(s_t|\xi_t,x')|p(s_t|\xi_t,x))}\right], \tag{14}$$

[8] The agent will loop back to $\mathcal{R}_1^n$, increases his chances of winning a high utility resources, instead of moving through a large number of undesirable resources.

where $\mathcal{N}'$ and $\mathcal{N}$ denote sets of agents containing and not containing x' correspondingly. In (11), we use the binomial expansion, in (12) the fact that the factors in front of the exponent do not depend on s_t.

For the inverse direction, $\mathcal{D}_{\lambda+1}(p(s_t|\xi_t,x)\|p(s_t|\xi_t,x'))$ changes to $\mathcal{D}_{\lambda}(p(s_t|\xi_t,x)\|p(s_t|\xi_t,x'))$, where the expectation is taken over $p(s_t|\xi_t,x)$. Then, since $f(x) = \frac{1}{x}$ is a convex function for $x > 0$, we can use the definition of convexity (more specifically, that $\frac{1}{(1-q)x+qy} \leq (1-q)\frac{1}{x} + q\frac{1}{y}$), and then apply the same steps as above:

$$\mathbb{E}_{s_t}\left[\left(\frac{p(s_t|\xi_t,\mathcal{N}')}{p(s_t|\xi_t,\mathcal{N})}\right)^{\lambda+1}\right] \tag{15}$$

$$= \mathbb{E}_{s_t}\left[\left(\frac{p(s_t|\xi_t,x)}{(1-q)p(s_t|\xi_t,x)+qp(s_t|\xi_t,x')}\right)^{\lambda}\right] \tag{16}$$

$$\leq \mathbb{E}_{s_t}\left[\left((1-q)+q\frac{p(s_t|\xi_t,x)}{p(s_t|\xi_t,x')}\right)^{\lambda}\right] \tag{17}$$

$$= \mathbb{E}_{k\sim\mathcal{B}(\lambda,q)}\left[\mathbb{E}_{s_t}\left[\left(\frac{p(s_t|\xi_t,x)}{p(s_t|\xi_t,x')}\right)^{k}\right]\right] \tag{18}$$

$$= \mathbb{E}_{k\sim\mathcal{B}(\lambda+1,q)}\left[e^{k\mathcal{D}_{k+1}(p(s_t|\xi_t,x)|p(s_t|\xi_t,x'))}\right]. \tag{19}$$

Using the above, we can compute the privacy cost $c_t(\cdot)$. □

E Paper Assignment

E.1 Setting

In this test-case, we protect the reviewers' preferences during the paper assignment phase of a conference. We used the multi-aspect review assignment evaluation dataset (Karimzadehgan and Zhai 2008). It contains 73 papers (which corresponds to the set of resources $\mathcal{R}$ in our setting) from the ACM SIGIR conference of 2007, and 189 prospective reviewers (which corresponds to the set of agents $\mathcal{N}$) composed by authors of published papers in the top information retrieval conferences between 1971-2006. Each paper and each reviewer is represented by a 25-dimensional binary label, representing one of the 25 major areas of ACM SIGIR (Karimzadehgan, Zhai, and Belford 2008).

We used the 25 major areas to define the privacy regions. Specifically, for each reviewer and paper, we selected uniformly at random one of the subject areas that they belong to, and set it as the *primary* subject area. The primary subject area is unique, and identifies the region. The proposed Piecewise Local Differential Privacy demands that users belonging to the same region be indistinguishable from the attacker's point of view. This would correspond to reviewers with the same primary subject area. We refer to the remaining subject areas as *secondary*. The maximum number of secondary subject areas of any adversary in a region defines the range of that region (reviewers are indistinguishable in that range). In this test-case, we consider adversaries with at most 2, 3, and 4 additional subject areas[9]. In layman's terms, a reviewer would be indistinguishable from any other reviewer that has the same primary subject area, and is an expert in at most 3, 4, and 5 areas in total.

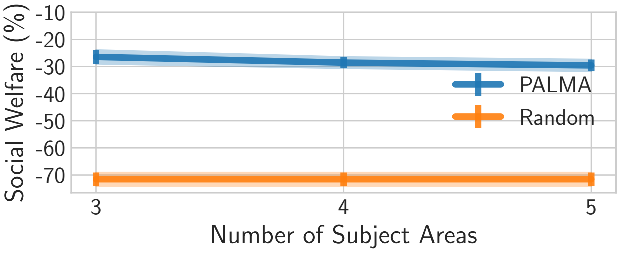

Figure 7: Loss in social welfare compared to the non-private, optimal solution for increasing size of the privacy region (i.e., number of subject areas).

Finally, for each paper and reviewer, we convert the 25-dimensional binary label to a continuous-valued vector. Specifically, the primary subject area is assigned the value 1, all the secondary subject areas are assigned the value 0.5, and the rest of the areas are assigned the value 0.1. The latter reflects the fact that conferences trust the expertise of reviewers to asses the quality of papers in a broader area. Following the literature (Ahmed, Dickerson, and Fuge 2017), we used the cosine similarity (Eq. 20) of their label vectors to compute the utility of a paper to a reviewer.

$$u_n(r) = \frac{\vec{n}\cdot\vec{r}}{\|\vec{n}\|\|\vec{r}\|} \tag{20}$$

where $\vec{n}$ ($\vec{r}$) denotes the 25-dimensional label of agent n (resource r).

Note that in a real-world paper assignment scenario, each reviewer would be required to review more than one paper (i.e., our matching graph would be a bipartite hypergraph). This can be easily handled by PALMA. Specifically, each reviewer will be represented by x 'copies', where x is the number of papers each reviewer should review. Then, a resource (paper) would only signal agent n that it is free (line 19 of Alg. 1) if (i) it has been assigned to less than y agents – where y represents the number of reviews per paper – and (ii) a 'copy' of agent n has not acquired the resource. Nevertheless, this is out of the scope of this paper; the goal of this test-case is to provide additional evidence on the performance of PALMA on real data. Thus, we opted to assign each reviewer to only one paper.

Finally, in this test-case, we set $q = 0.05$.

E.2 Simulation Results

Social Welfare PALMA loses $-26.4 \pm 2.5\%$ to $-29.6 \pm 2.2\%$ in social welfare compared to the non-private, optimal solution (Figure 7). As a baseline, we have plotted the maximally private solution (i.e., the centralized random), which losses $71.5 \pm 2.5\%$.

Privacy Figure 8 depicts the maximum (out of all the 16 runs) and median (average median value over the 16 runs) per-agent ε for increasing values of the size of the privacy region (i.e., number of additional subject areas). The average

[9] This would correspond to cosine distance of ≤ 0.2, ≤ 0.25, and ≤ 0.3, respectively, from an agent that has a single subject area; the primary subject area of the corresponding region

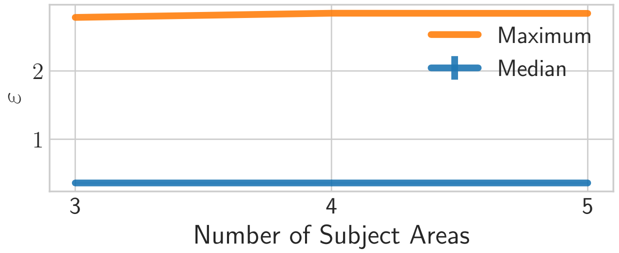

Figure 8: Maximum (orange line) and median (blue line) per-agent ε for increasing values of the size of the privacy region (i.e., number of subject areas). The shaded area represents the range between the maximum and minimum value of the median.

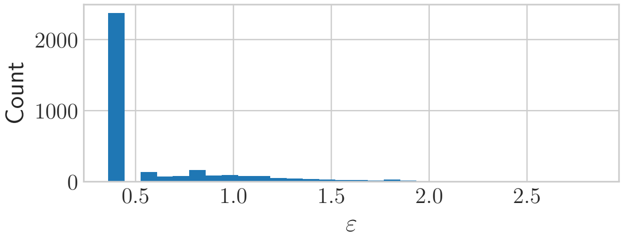

Figure 9: Histogram of per-agent ε for a privacy region with 3, 4, or 5 subject areas. We include all 16 runs (16 (runs) $\times$ 73 (agents matched with a resource) $\times$ 3 (test-cases with different number of subject areas) = 3504 data points in total).

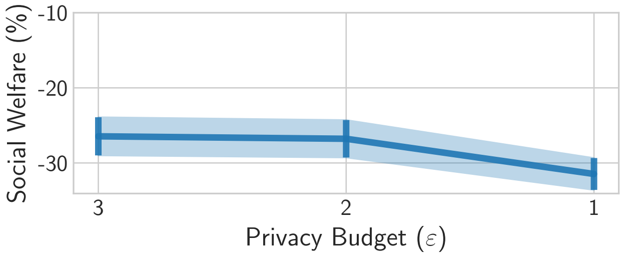

Figure 10: Loss in social welfare for decreasing privacy budget (ε) for a privacy region with 3 subject areas.

value of the median is 0.36. The maximum per-agent ε is bounded by the privacy budget (i.e., $\varepsilon = 3$).

Figure 9 plots the histogram of the per-agent ε for a privacy region with 3 subject areas. Only 15% of the agents have $\varepsilon > 1$, and only 1.3% have $\varepsilon > 2$. The vast majority of the agents 67.7% have $\varepsilon \leq 0.5$, which means that they benefit from really high degree of privacy.

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 budget (i.e., higher level of privacy) to the solution quality. Figure 6 depicts the achieved social welfare for decreasing privacy budget (ε). Restricting the budget from $\varepsilon = 3$ to $\varepsilon = 2$ results in only $\approx 0.3\%$ additional loss in social welfare, and from $\varepsilon = 3$ to $\varepsilon = 1$ in only $\approx 5\%$ additional loss.

F Societal Impact

The rapid proliferation of intelligent systems and autonomous agents has the potential to positively impact many facets of our daily lives. However, harnessing their power requires massive amounts of personal data to be collected, stored, processed, and analyzed – often by resource-constrained devices. The latter has raised serious privacy concerns and has resulted in an accelerated growth of privacy advocacy movements. Our work shows that harnessing the potential of intelligent systems does not have to compromise privacy.

We provide a *practical* and *applicable* framework – PALMA can run *on-device* – for solving one of the fundamental problems of multi-agent systems (finding matches, and allocations), while providing *strong* worse-case *privacy* guarantees.

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.

Ahmed, F.; Dickerson, J. P.; and Fuge, M. 2017. Diverse Weighted Bipartite B-Matching. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, 35–41. AAAI Press. ISBN 9780999241103.

Alonso-Mora, J.; Samaranayake, S.; Wallar, A.; Frazzoli, E.; and Rus, D. 2017. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proc. of the National Academy of Sciences* .

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. ISBN 9781450324779. doi:10.1145/2508859.2516735. URL https://doi.org/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, B.; Thakkar, O. D.; and Thakurta, A. 2020. Privacy Amplification via Random Check-Ins. In *Advances in Neural Information Processing Systems 33: NeurIPS 2020.* URL https://proceedings.neurips.cc/paper/2020/hash/313f422ac583444ba6045cd122653b0e-Abstract.html.

Chatzikokolakis, K.; Andrés, M. E.; Bordenabe, N. E.; and Palamidessi, C. 2013. Broadening the Scope of Differential Privacy Using Metrics. In *Privacy Enhancing Technologies*. 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*.

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 *Proc. of the 17th International Conference on Autonomous Agents and MultiAgent Systems*.

Gil, M.; Alajaji, F.; and Linder, T. 2013. Rényi divergence measures for commonly used univariate continuous distributions. *Information Sciences* 249: 124–131.

Hsu, J.; Huang, Z.; Roth, A.; Roughgarden, T.; and Wu, Z. S. 2014. Private Matchings and Allocations. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '14, 21–30. New York, NY, USA: Association for Computing Machinery. ISBN 9781450327107. doi:10.1145/2591796.2591826. URL https://doi.org/10.1145/2591796.2591826.

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)*.

Karimzadehgan, M.; and Zhai, C. 2008. Data Set for Multi-Aspect Review Assignment Evaluation. http://sifaka.cs.uiuc.edu/ir/data/review.html. Accessed: 2021-01-14.

Karimzadehgan, M.; Zhai, C.; and Belford, G. 2008. Multi-Aspect Expertise Matching for Review Assignment. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*. ISBN 9781595939913. doi:10.1145/1458082.1458230. URL https://doi.org/10.1145/1458082.1458230.

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

Lovász, L.; and Plummer, M. D. 2009. *Matching theory*. 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.

Riley, C.; van Hentenryck, P.; and Yuan, E. 2020. Real-Time Dispatching of Large-Scale Ride-Sharing Systems: Integrating Optimization, Machine Learning, and Model Predictive Control. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*.

Roth, A. E.; Sönmez, T.; and Ünver, 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 *IEEE International Conference on Big Data (Big Data)*. 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*.

Van Erven, T.; and Harremos, P. 2014. Rényi divergence and Kullback-Leibler divergence. *IEEE Transactions on Information Theory* 60(7): 3797–3820.

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.