A Distributed Differentially Private Algorithm for Resource Allocation in Unboundedly Large Settings

Panayiotis Danassis
École Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland
panayiotis.danassis@epfl.ch

Aleksei Triastcyn
École Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland
aleksey.tryastsyn@alumni.epfl.ch

Boi Faltings
École Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland
boi.faltings@epfl.ch

ABSTRACT

We introduce a *practical* and *scalable* algorithm (PALMA) for solving one of the fundamental problems of multi-agent systems – finding matches and allocations – in *unboundedly large* settings (e.g., resource allocation in urban environments, mobility-on-demand systems, etc.), while providing *strong worst-case privacy* guarantees. PALMA is decentralized, runs on-device, requires no inter-agent communication, and converges in constant time under reasonable assumptions. We evaluate PALMA in a mobility-on-demand and a paper assignment scenario, using *real data* in both, and demonstrate that it provides a strong level of privacy ($\varepsilon \leq 1$ and median as low as $\varepsilon = 0.5$ across agents) and high-quality matchings (up to 86% of the non-private optimal, outperforming even the privacy-preserving centralized maximum-weight matching baseline).

KEYWORDS

Resource Allocation; Coordination and Cooperation; Differential Privacy; Maximum-weight Matching; Weighted Matching; Assignment Problem; Decentralized; On-device

ACM Reference Format:
Panayiotis Danassis, Aleksei Triastcyn, and Boi Faltings. 2022. A Distributed Differentially Private Algorithm for Resource Allocation in Unboundedly Large Settings. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022,* IFAAMAS, 9 pages.

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 [8] to kidney exchange [24] – can be formulated and solved as a weighted 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 [25]), 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 algorithm designed to tackle *all* of the aforementioned challenges.

PALMA is a privacy-preserving adaptation of ALMA [6, 7, 10]; a recently proposed algorithm for real-world, large-scale applications that solves the first two challenges. As such, it 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) [11] (and its variants) has emerged as the de facto standard for protecting the privacy of individuals. Informally, a DP algorithm ensures indistinguishability on the output distributions for any neighboring inputs. We have designed a defense mechanism for PALMA based on the idea of randomized response [33] – which involves adding controlled randomness – that results in indistinguishability under Local DP [15].

One final challenge arises when it comes to *large-scale* multi-agent systems with a *diverse* set of agents, as it is hard to achieve a meaningful privacy guarantee – in a practical way – using standard (L)DP if the problem has a large output space (e.g., matches, and allocations) [17, 29]. Conventional (L)DP mechanisms often require adding a lot of random noise to achieve a meaningful privacy guarantee, which in turn leads to a 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 protects all users – independent of their characteristics – by the same guarantee. 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 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 because it is likely public or easily available and, thus, does not need to be hidden (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 (e.g., a taxi in Manhattan will not be assigned to serve a request in Brooklyn, and an expert on auctions would not be assigned to review a robotics paper, thus, there is no need for indistinguishably between taxis in different boroughs or reviewers on different fields).

To solve this challenge, we motivate and develop a ‘context-aware’ 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

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online.

distance; agents with utility functions that have images close in distance to each other would be indistinguishable from the attacker's point of view.

1.1 Our Contributions

(1) We propose *PALMA*, the *first* practical and scalable privacy preserving algorithm for weighted matching in *unboundedly large* settings with thousands of agents (e.g., resource allocation in urban environments, intelligent infrastructure, IoT devices, etc.).

(2) 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.*

(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, $\varepsilon \leq 1$ and a median value as low as 0.5 across agents for $\delta = 10^{-5}$, and matchings of high quality (up to 86% of the non-private optimal).

The decentralized algorithm and corresponding privacy definition allows PALMA to adapt the noise added for obfuscation to the privacy budget of each agent. This achieves significantly better performance than the centralized Hungarian algorithm with the fixed obfuscation required to achieve the same privacy guarantees with an untrusted server.

1.2 Related Work

Finding a maximum-weight matching is one of the best-studied combinatorial optimization problems [20, 26]. Yet, 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 [25], 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 [7]. The proposed approach, PALMA, *preserves* the aforementioned two properties of ALMA, thus, dealing with the first two of the posed challenges.

Differential Privacy (DP) [11–14] has emerged as the de facto standard for protecting the privacy of individuals (see the supplement [9] for the definition of DP, along with intuitive examples)[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) [15]. LDP is a generalization of DP that provides a bound on the 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 the decline of the solution quality. In fact, it is impossible to have both meaningful social welfare and privacy guarantees in matching problems under (L)DP [17]. (L)DP ignores specifics of AI applications, such as a focus on a given task or a particular data distribution.

Our work is inspired by the literature on 'data-aware' privacy notions [30, 32] and distance-based generalisations of DP [3, 5]. As a matter of fact, there are works that utilize such distance-based notions to solve a weighted matching problem in specific domains (e.g., [16, 22]). Yet, these are centralised approaches and thus face a (computation and communication) complexity barrier (refer back to the aforementioned challenges (i) and (ii) of real-world matching problems). ALMA can also be combined with other existing notions of privacy (e.g., LDP or geo-indistinguishability [3]), yet the solution quality is inferior compared to the proposed, carefully crafted (with ALMA in mind) noise, as we demonstrate in our evaluation.

2 PIECEWISE LOCAL DIFFERENTIAL PRIVACY (PLDP)

Inspired by the notions of Bayesian DP [31] – 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 [5] and geo-indistinguishability [3] – 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.

Let $\mathcal{M} : \mathcal{D} \to \mathcal{A}$ be a randomized function with domain $\mathcal{D}$ and range $\mathcal{A}$. In the context of matching problems in multi-agent systems, $\mathcal{D}$ is the space of 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 $\mathcal{S} \subset \mathcal{A}$ it holds:*

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

2.1 Motivation

Consider a mobility-on-demand (MoD) application (e.g., ridesharing). A MoD company can operate across multiple cities, countries, or even continents. If a MoD provider employs traditional DP (e.g., LDP) to protect all users (independently of their characteristics)

[1] For a more comprehensive overview, we refer the reader to [15, 30].

with the same guarantee, the achieved social welfare will be as good as a *random solution*[2] in large-scale environments. This is because the *support* of any agent has to include *every resource* (otherwise an adversary could distinguish between agents), i.e., a request in Manhattan might be paired with a taxi in Brooklyn. Moreover, it is reasonable to assume an informed attacker (e.g., one that knows the city of residence), and users may be willing to reveal approximate location information (it is most likely acceptable to disclose the fact that an individual is in Manhattan, however disclosing the exact location 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) that would exclude infeasible matches.

The rationale behind PLDP 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. In other words, while an agent can be distinguished from agents outside his zip code, he is still indistinguishable from *all* agents inside his 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 sub-domains $\mathcal{D}_i$. 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* [1, 15], will also hold for any pair x, x' from a given $\mathcal{D}_i$, as long as these points do not dynamically change sub-domains 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 (vs. Distance-based Generalisations of DP)

PLDP closely resembles another well-known privacy notion, geo-indistinguishability [3], which is based on a generalization of DP [5]. 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, there is also a crucial upside to this subtle difference 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 [1]. 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 $O(\sqrt{T})$ [1].

A second advantage of PLDP is that, contrary to geo- indistinguishability, it does not require a metric space (i.e., a natural ordering). As an example, this makes PLDP easier to apply in settings like our paper assignment application where each agent/resource is represented by a 25-dimensional binary label (see supp.). In this example, there is ordering in each dimension, but not across them.

3 PALMA: A PRIVACY-PRESERVING WEIGHTED MATCHING ALGORITHM

3.1 The Assignment Problem

The assignment problem refers to finding a maximum-weight matching in a weighted bipartite graph[3], $\mathcal{G} = \{\mathcal{N} \cup \mathcal{R}, \mathcal{E}\}$. 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.

For simplicity, in the rest of the paper we assume $N = R$. This is *not required* by PALMA (or 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, see [7]).

3.2 Learning Rule

We assume each agent is interested in (potentially) a subset of the total resources $Q^n \subseteq \mathcal{R}$. 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. Let g denote the agent's strategy. PALMA is run *independently and in parallel by all the agents.* Each agent 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[4], 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 can be found in Alg. 1.

3.2.1 *Resource Selection Distribution.*

In the original ALMA, each agent sorts the resources in decreasing order of utility $(r_1, \dots, r_R)$.

[2] The solution that results of picking edges randomly in a fully connected bipartite graph containing all agents and resources.

[3] ALMA (and thus PALMA) can be applied in general graphs as well (see [8]).

[4] 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).

Table 1: Nomenclature, Algorithm 1

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

Algorithm 1 PALMA: Privacy-preserving ALtruistic MAtching.

```
1: **Initialize** $s \leftarrow 1$, $g \sim P_S^n(\cdot)$, $c \leftarrow 0$, $converged \leftarrow False$
2: **Calculate** $c_{max}$ according to Equation 6
3: **procedure** PALMA
4:     **while** !converged **do**
5:         **if** $g = A_r$ **then**
6:             Agent $n$ attempts to acquire $r$
7:             **if** Collision($r$) **then**
8:                 **if** $c + c_{max} \leq B_n$ **then**
9:                     Back-off (set $g \leftarrow Y$) with probability $P_B^n(\cdot)$
10:                    $c \leftarrow c + c_{max}$
11:                **else**
12:                    Back-off (set $g \leftarrow Y$) with prob. $B_{\text{Noise}}(\cdot)$
13:            **else**
14:                converged $\leftarrow True$
15:        **else** ($g = Y$)
16:            $s \leftarrow (s + 1) \mod R$
17:            **if** $c + c_{max} \leq B_n$ **then**
18:                Agent $n$ monitors $r \sim P_S^n(\cdot)$
19:                $c \leftarrow c + c_{max}$
20:            **else**
21:                Agent $n$ monitors $r \sim S_{\text{Noise}}(\cdot)$
22:            **if** Free($r$) **then** set $g \leftarrow A_r$
23: **Output** $r$, such that $g = A_r$, and $(\varepsilon, \delta) \leftarrow$ getPrivacy($c$) (Eq.4.1)
```

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 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., $\mathcal{N}^n = \{n' : u_{n'}(\cdot) \in \mathcal{D}_i \wedge u_n(\cdot) \in \mathcal{D}_j \implies i = j\}$. 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, each agent n independently generates the sets $(\mathcal{R}_1^n, \ldots, \mathcal{R}_i^n, \ldots, \mathcal{R}_R^n)$, 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 18 of Algorithm 1):

$$P_S^n(i, s, \zeta_S) = \zeta_S \times P_{\text{WaR}}(i, s, n) + (1 - \zeta_S) \times S_{\text{Noise}}(i, s, n^*) \quad (1)$$

$$P_{\text{WaR}}(i, s, n) = \frac{u_n(r_i)}{\sum_{r \in \mathcal{R}_s^n} u_n(r)} \quad (2)$$

Equation 1 defines a mixture distribution, composed of (a) selecting in a weighted at random fashion using the utilities of agent n ($P_{\text{WaR}}(i, s, n)$, given by Equation 2), and (b) a distribution that introduces noise ($S_{\text{Noise}}(\cdot)$) to the selection process. ζ_S tunes the magnitude of the introduced randomness.

The introduced noise can be any distribution that is known and common for all agents (can be domain specific). For example, it could simply be a uniformly at random selection in the set of resources $\mathcal{R}_s^n$. In this work, we take advantage of domain knowledge. Specifically, let n^* denote a 'representative' agent of the Neighborhood of agent n. This can be for example a(n) (potential) agent located in the center of the neighborhood in a mobility-on-demand application. Then, the common distribution (i.e., noise) can be to play in a uniformly at random manner according to the utility function of the representative agent, i.e., $S_{\text{Noise}}(i, s, n^*) = P_{\text{WaR}}(i, s, n^*)$. In section 3.2.3, we provide a concrete example on the fragmentation of the utility space into neighborhoods, the representative agent, and the selection and back-off probabilities.

3.2.2 ***Back-off Distribution***. The back-off probability, $P_B^n(\cdot)$ (line 9 of Algorithm 1), is computed individually and locally based on each agent's expected utility loss that he will incur if he switches:

$$loss(i, s, n) = u_n(r_i) - \sum_{r_j \in \mathcal{R}_{s+1}^n} \frac{u_n(r_j)}{\sum_{r \in \mathcal{R}_{s+1}^n} u_n(r)} u_n(r_j) \quad (3)$$

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

$$f(loss) = \begin{cases} 1 - \gamma, & \text{if } loss \leq \gamma \\ \gamma, & \text{if } 1 - loss \leq \gamma \\ 1 - loss, & \text{otherwise} \end{cases} \quad (4)$$

where γ places a threshold on the minimum / maximum back-off probability. 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 18 of Algorithm 1). Finally, $P_B^n(\cdot)$ is given by Equation 5:

$$P_B^n(i, s, \zeta_B) = \zeta_B \times f(loss(i, s, n)) + (1 - \zeta_B) \times B_{\text{Noise}}(i, s, n^*) \quad (5)$$

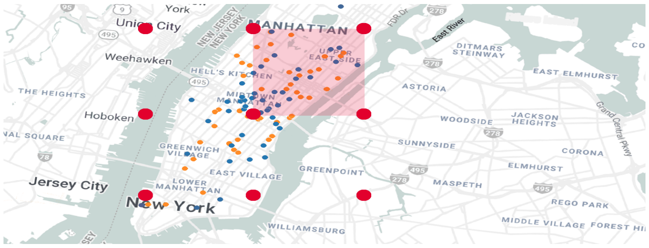


Figure 1: A visual representation of the regions ($\{\mathcal{D}_i\}$) of PLDP for the mobility-on-demand application. Red dots denote the edge points of each region ($\ell = 4000$). Orange dots represent the agents (requests), and blue dots represent the resources (vehicles) in our dataset. As an example, an agent in the overlaid rectangle could be located *anywhere* in the rectangle from the attacker's point of view.

The back-off distribution is mixture between acting according to an agent's own utility function ($f(loss(i, s, n))$), and a distribution that introduces noise ($B_{\text{Noise}}(\cdot)$) to the back-off process. ζ_B tunes the magnitude of the introduced randomness. As was the case with the selection distribution, the introduced noise for the back-off distribution can be any distribution that is known and common for all agents. In this work, we set $B_{\text{Noise}}(i, s, n^*) = f(loss(i, s, n^*))$, i.e., the 'noise' distribution refers to backing-off according to the utility function of the 'representative' agent (described in Section 3.2.1).

3.2.3 ***Elaborative Example on Neighborhoods.*** In what follows, along with Section 4.1.1, we will provide an elaborative, practical example of the key notions of PALMA.

PLDP is used to protect the utility function of agents. Consider the space of all possible utility functions, and then consider the space of the images of those utility functions. We fragment the former into sub-spaces D_i, such that for two utility functions that belong to the same D_i, their image is 'close' in distance. In simple terms this means that the actual utility value of a resource would be similar for agents with utility functions in the same sub-space D_i. The fragmentation is performed by $\phi(\cdot)$.

Each agents selects his own $\phi(\cdot)$ based on his privacy needs. The choice of $\phi(\cdot)$ is public information. For simplicity, in this work, we assume that every agent has the same $\phi(\cdot)$. The choice of $\phi(\cdot)$ fragments the space of agents into regions; the image of the utility function of every agent in a region is close in distance to every other agent in the same region. The definition of the region ($\phi(\cdot)$) as well as the distance metric are domain specific.

As a concrete example, consider a mobility-on-demand (MoD) application (e.g., ridesharing). Let the utility of each agent (ridesharing user) be inversely proportional to the distance (in meters) from the resource (vehicle). In this case, we can split the are of operation into rectangular regions, as shown in Figure 1; agents in the same region would have similar utilities for each resource.[5]

To compute his neighbors, an agent considers *every possible agent* that could belong in his region, regardless if this agent exists. Expanding on our MoD example, we can consider having an agent ridesharing user) every, e.g., 10m on the map. In a 10^6m^2 region, the neighborhood will include 10^4 agents. Each of these agents has his own preference (ordering) of resources. Using these preferences, we can construct the sets $\mathcal{R}^n_1, \dots, \mathcal{R}^n_R$, where the set $\mathcal{R}^n_i$ contains the i^{th} most preferred resource of each neighbor. The construction of the neighborhoods needs to be performed once, offline. PLDP guarantees that each agent is *indistinguishable from all his neighbors* (i.e., every potential agent that could exist in his region) from the attacker's point of view.

Finally, the 'representative' agent of each region can be a 'virtual' agent located at the center of the region. Given that $\phi(\cdot)$ is public – and thus the fragmentation into regions as well – the selection and back-off distribution of the representative agent is also public and common for all agents.

3.3 Communication and Computation Complexity

PALMA (just like ALMA [7]) does not require any inter-agent communication[4]. The initialization is linear to the size of the region, $O(\max_i |\mathcal{D}_i|)$, but this can be done once off-line. The accounting of the privacy loss is $O(1)$. Finally, PALMA converges in polynomial time in the general case, and in *constant* time in the realistic case where 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 ($\mathcal{V}^r \subset \mathcal{N}$) (see supplement [9]).

3.4 Privacy Mechanism

PALMA's defense mechanism is based on the idea of randomized response [33], and involves adding controlled randomness in (i) the resource selection and (ii) back-offs, parametrized by ζ_S and ζ_B, respectively (see Equation 1 and 5). The idea is that the agent first flips a coin to decide whether to act truthfully. Then, with probability ζ_S (or ζ_B), the agent plays according to its true selection (or back-off) function; with probability $1 - \zeta_S$ (or $1 - \zeta_B$), the agent plays according to a public, common distribution.

Moreover, each agent has a privacy budget of $\varepsilon = B_n$. Upon depletion in the course of using the above mechanisms (see lines 8 & 17 of Algorithm 1), the agent will play *noisy* actions (see lines 12 & 21 of Algorithm 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., [15]). We employ the accounting framework introduced in [32] 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 [1] for the subsampled Gaussian mechanism and Rényi accountant [21]. 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 $\sigma_{1:T}$ ($\xi_{1:T}$).

[5] Note that we protect the privacy of the agents, not the resources; thus, the resources (vehicles) do not need to belong to any region, and can be matched with any agent regardless of his region.

In the context of PALMA, these signals represent either an attempt to acquire a resource, or a back-off from a previously contested resource[6], while the auxiliary information corresponds to s (which determines the set of resources $\mathcal{R}_s$, see Equation 1, 5). Following [32], we also introduce the notion of *privacy cost*:

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

where $\mathcal{D}_\lambda(\cdot\|\cdot)$ is the Rényi divergence of order λ (see [9]).

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:

$$c_{max} = \max \begin{cases} \max\limits_{\xi_t \in \{1,\dots,R\}} \max\limits_{x' \in \mathcal{N}^x} \max\limits_{\sigma_t \in \mathcal{R}^x_{\xi_t} \sim P^n_S(\cdot)} c_t(\cdot) \\ \max\limits_{\xi_t \in \{1,\dots,R\}} \max\limits_{x' \in \mathcal{N}^x} \max\limits_{\sigma_t \in \mathcal{R}^x_{\xi_t} \sim P^n_B(\cdot)} c_t(\cdot) \end{cases} \tag{6}$$

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 according to the noise distributions. Agents can then adjust their privacy parameters accordingly. The actual privacy loss is accounted on the fly during execution (see lines 10 and 19 of Algorithm 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 Bayesian accountant [32], the moments accountant [1] and the Rényi accountant [21] are equivalent in this case, resulting in:

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

It is important to note that the above ε and δ should not be published, since the agent uses his own utility function to calculate the cost (in Equation 6).

4.1.1 *Elaborative Example on the Privacy Cost Calculation*.

In this section we expand on our practical example on MoD systems introduced in Section 3.2.3.

Recall that PLDP provides *Local* DP guarantee, meaning a bound on the outcome probabilities for *any* pair of individual agents, inside the region. As such, to compute the privacy cost per round, each agent n has to identify the neighbors that would result to the maximum privacy loss (i.e., their selection (back-off) distributions result in the largest Rényi divergence, see Equation 6). Thus, each agent n independently identifies two agents n', and n'' *from his neighborhood* that result in the worst privacy loss given the agent's selection and back-off distributions (Equation 1 and 5, respectively). Then, he can compute the *worst case* privacy loss in any round by taking the maximum of the two values (Equation 6). Using this information, each agent is able to (i) compute his total privacy cost *a priori* and adjust his privacy parameters accordingly, (ii) keep track of his privacy budget at every time-step, and (iii) calculate his total ε after convergence. This process needs to happen *once*, *offline*. As mentioned, each agent can adjust the size of the neighborhood $\mathcal{N}^n$ (e.g., length ℓ, see Section 6.2) according to his privacy needs.

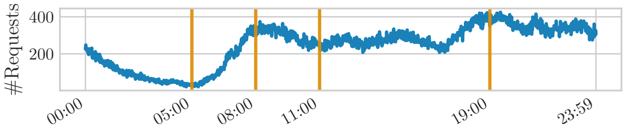


Figure 2: Request per minute in Manhattan on Jan. 15, 2016. Vertical lines denote the selected evaluation instances.

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 $\delta = 10^{-5}$). Each problem instance is run 32 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 $\zeta_S = 0.2$, $\zeta_B = \gamma = 0.05$, $B_n = 1$, $\lambda = 32$.

6 TEST-CASE 1: MOBILITY-ON-DEMAND

6.1 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 raises privacy concerns. Such a problem is ideal to showcase PALMA, as explained in Section 2.1. Moreover, contrary to other approaches (e.g., [16, 22]) PALMA is *decentralized* and employs *Local DP*, providing privacy against a malicious data curator.

The Ridesharing and Fleet Relocation problem can be decomposed into three weighted matching sub-problems, all of which can be solved efficiently by ALMA [8] (and thus by PALMA as well). In this test-case we will focus on passenger to vehicle matching, using PLDP and PALMA to provide a *scalable*, *on-device*, decentralized solution that *protects user preferences* (user location in this context).

6.2 Setting

Our evaluation setting is specifically designed to *resemble reality as closely as possible*, following the modeling of [8]. We have used the NYC yellow taxi trip records [28]. For every request, the dataset provides amongst others the geo-location coordinates.

We report results on four 30s instances on a typical day (Jan 15th). These instances were selected to represent various distributions of demand (see Figure 2): the two highest peaks, the lowest peak, and a mid-day low[7]. We selected 30s periods because in practice the granularity of in-batches approaches for MoD services is between[8]

[6] In an arbitrary domain, the signal would correspond to an action of an agent.

[7] Specifically, 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.

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

10s to 30s [2, 8, 22, 23]. It is important to stress this *does not affect the scalability* of the proposed approach. Running PALMA for a day, for example, would simply result in running 24 × 60 × 2 batches (as was done in [8]). Assuming similar distributions for requests and vehicles[9], the social welfare and privacy cost of each agent will remain approximately the *same*, since the privacy cost (Equation 6) *only depends on the size of the region* $\mathcal{D}_i$.

The set of agents $\mathcal{N}$ is composed by the requests in Manhattan (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[10] to calculate the distance in each coordinate), as it has been found to be a close approximation of the actual driving distance in Manhattan [8]. 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). We opted to use an exponential function to enable short pick-up times, as research conducted by ridesharing companies shows that a short pick-up time is important for passengers' satisfaction [4, 27].

The map is divided into *fixed* square regions of edge length ℓ (which correspond to the $\mathcal{D}_i$). PLDP demands that a user is indistinguishable, from the attacker's point of view, from any potential user that could exist in the same region[11] (i.e., all his neighbors, see Sections 3.2.1 and 3.2.3). We have evaluated $\ell \in \{1000, 2000, 3000, 4000\}$ m, which roughly correspond to an area of $\{45.6, 182.5, 410.5, 730\}$ city blocks[12]. Figure 1 offers a visual representation of the setting.

6.3 Baselines

We employ the centralized Hungarian algorithm [19] to compute the non-private maximum-weight – i.e., optimal in terms of social welfare – solution, which we use to compare the loss in social welfare of all of the evaluated algorithms. We compare PALMA against three privacy-preserving baselines:

1. The Hungarian algorithm [19] – which is an optimal assignment centralized algorithm – made private by obfuscating (adding noise) the geo-location coordinates according to geo-indistinguishability [3] (similarly to [22]).
2. The original ALMA [7] under similarly obfuscated (noisy) geo-location coordinates[13].
3. The maximally private solution (i.e., the centralized random).

For the geo-indistinguishability-based baselines, we calculated a noisy geo-location for each agent and resource, according to Algorithm 2, which can be found in the supplement [9].

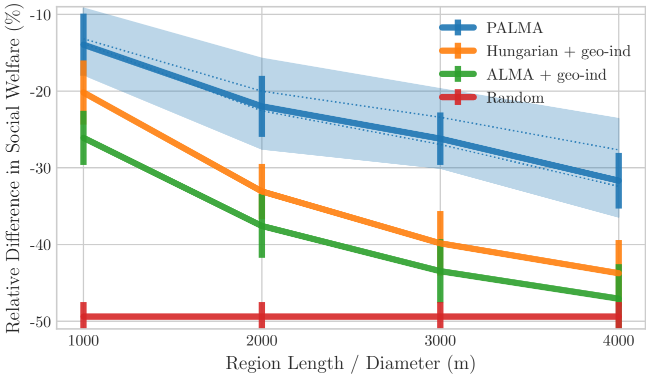


Figure 3: Loss in SW compared to the non-private, optimal solution for increasing region edge length (ℓ) and $\varepsilon = 1$. The dotted lines represent the upper ($\varepsilon \to \infty$) and lower ($\varepsilon = 0$) bound for PALMA, while the shaded area adds one standard deviation to the aforementioned bounds (see Section 6.4).

6.4 Simulation Results: Social Welfare

For $\varepsilon = B_n = 1$ given $\delta = 10^{-5}$ (Figure 3), PALMA loses between 13.9 ± 4.1% ($\ell = 1000$) to 31.7 ± 3.6% ($\ell = 4000$) in social welfare compared to the non-private, optimal solution. The dotted lines represent the upper and lower bound; the upper bound assumes infinite budget ($B_n \to \infty$) thus the agents play according to their own utilities ($\zeta_S = \zeta_B = 1$), while the lower bound assumes zero budget ($B_n = 0$) thus the agents play according to the noise distribution ($\zeta_S = \zeta_B = 0$), i.e., according to the utilities of the representative agent. The shaded area adds one standard deviation to the aforementioned bounds.

For the same ε guarantee and the same length as the privacy diameter, Hungarian + geo-ind loses between 20.2 ± 4.2% to 43.7 ± 4.3%, while ALMA + geo-ind loses between 26.1±3.5% to 47.1±4.5%. Finally, the maximally private solution (i.e., the centralized random), losses 49.4 ± 2%.

PLDP and the carefully crafted noise of PALMA, allows PALMA to *outperform even the centralized optimal* solution (Hungarian + geo-ind) by 27.6% ($\ell = 4000$) to 30.9% ($\ell = 1000$). In fact, if we increase the privacy requirement to $\varepsilon = 0.75$, the improvement increases to 31.3% ($\ell = 4000$) to 45.9% ($\ell = 1000$). Note that, besides the higher social welfare for the same privacy guarantee, PALMA is inherently decentralized and orders of magnitude faster than the Hungarian.

6.5 Simulation Results: Privacy

While the worst-case guarantee is the same across the evaluated methods, PALMA yields a stronger result on a per-agent basis. In PALMA, every agent has a budget $\varepsilon = B_n$ and can compute a priori the maximum number of rounds until the budget is exhausted and he has to play according to the noise distributions (see Section 4.1). During runtime, though, most agents converge in a few rounds (i.e, few privacy mechanism invocations), thus accumulating smaller privacy loss compared to geo-ind based methods.

To demonstrate the latter, Figure 4 depicts the maximum (out of all the 32 runs) and median (average median value over the 32

[9] A reasonable assumption given that our choice of evaluated distributions covers all the extremes, and a typical mid-day demand.
[10] https://en.wikipedia.org/wiki/Haversine_formula
[11] We assume that potential neighbors are 100m apart in every direction.
[12] The standard city block in Manhattan is about 80 m × 274 m (https://en.wikipedia.org/wiki/City_block).
[13] Note that we also attempted to use the original ALMA with Local Differential Privacy, yet, due to the large problem size, the privacy budget only sufficed for one round.

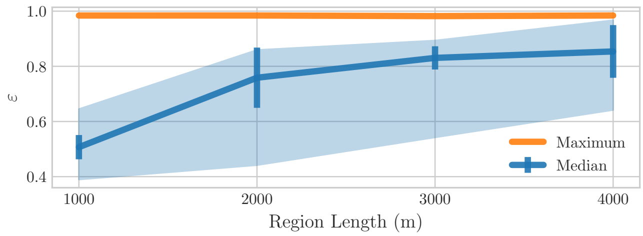
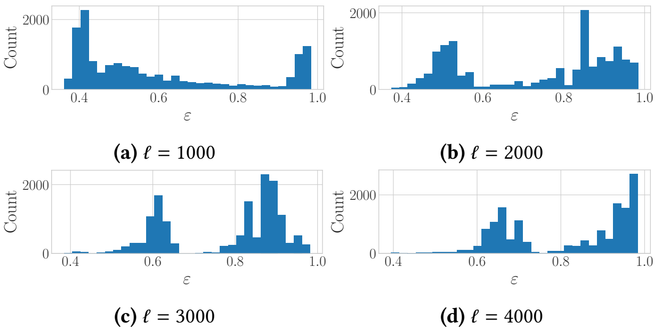


Figure 4: Maximum (orange) and median (blue) per-agent ε for increasing region length (ℓ). The shaded area represents the range between the max and min value of the median.



(a) ℓ = 1000

(b) ℓ = 2000

(c) ℓ = 3000

(d) ℓ = 4000

Figure 5: Histogram of per-agent ε for varying privacy region edge length. We include all 32 runs (32 (runs) × (17 + 154 + 116 + 174) (agents) = 14752 data points).

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 for $\ell = 1000$ is only 0.5. Of course, the maximum per-agent ε is bounded by the privacy budget (i.e., $\varepsilon = 1$). 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 5 plots the histogram of the per-agent ε for varying privacy region edge length (ℓ). For $\ell = 1000$ (Figure 5a), only 3572 out of 14752 agents (24.2%) have $\varepsilon > 0.75$. This is because the majority of the agents converge fast [7], thus only a small percentage of them exhaust their budget. In fact, almost half of the total agents (6759 / 14752, or 45.8%) have $\varepsilon \leq 0.5$. It is clear that the vast majority of agents benefit from really high degree of privacy.

6.6 Regions, Representative Agents, and Noise

In addition to the advantages of PLDP described in Section 2.3, there is another, more practical advantage that stems from the use of domain knowledge. The fragmentation function $\phi(\cdot)$ and the choice of the representative agent per region are domain specific. If the problem at hand (and by extension the utility function of the participating agents) is such that the representative agent has similar utilities to other agents in the region (and if we properly select the correct representative agent so that he is indicative of the agents in the region), then the social welfare will not degrade much, even under really strict budgets. Acting according to the representative agent, in such cases, allows for more informed allocations. This is a fundamental difference compared to, e.g., geo-indistinguishability, where the social welfare degrades in a significantly higher rate (as demonstrated in Figure 3). The latter can also be important for outlier agents, whose privacy cost per round might be high and thus lack the budget to play according to their own utilities for many rounds.

Regarding the choice of the fragmentation function $\phi(\cdot)$, there is a clear trade-off between the region size and the privacy cost per round, which in turn informs the amount of noise (ζ_S and ζ_B). Restricting our privacy guarantees to a region helps reduce the required noise, since all the agents in a region have similar preferences (less noise is needed to become indistinguishable). If the privacy cost per round is small, an agent can afford lower noise (larger ζ_S and ζ_B). Alternatively, acting according to the utilities of a properly chosen representative agent will still result in high quality allocations (especially in smaller regions, e.g., $\ell = 1000$), thus an agent might choose to accept higher noise in order to end up with much lower privacy cost at the end.

Finally, while in this work ζ_S and ζ_B are the same for all agents (see Section 5), one can potentially achieve better results using adaptive noise. For example, agents can assume lower noise for the first few time-steps, and gradually increase it over time. Note, that the noise selection scheme must not depend on the agents' preferences. We leave this open for future work.

7 TEST-CASE 2: PAPER ASSIGNMENT

We ran a second test-case (see supplement [9]), where we use PLDP and PALMA to protect the reviewers' preferences during the paper assignment phase of a conference, using real data form [18]. PALMA achieved similar results: loss in social welfare < 22% (the maximally private solution loses 71.5%); $\varepsilon \leq 1$ and a median value of 0.36.

8 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. We propose *PALMA*, a practical and scalable privacy-preserving algorithm for weighted matching along with PLDP, a 'context-aware' privacy model that takes into account the 'distance' between two utility functions. This ensures indistinguishability between agents with similar preferences. PALMA is decentralized, runs on-device, requires no inter-agent communication, converges in constant time under reasonable assumptions, and provides a *strong level of privacy* ($\varepsilon \leq 1$ and median as low as $= 0.5$), while achieving high quality matchings (up to 86% of the non-private optimal). To the best of our knowledge, we are the first to develop a practical and scalable framework for weighted matching and resource allocation in general, unboundedly large, multi-agent systems.

Supplementary Material. Please see [9] for an intuitive explanation of Differential Privacy with examples, implementation and complexity details, additional simulation results, and more.

Acknowledgements. This research was partially supported by TAILOR, a project funded by EU Horizon 2020 research and innovation programme under GA No 952215

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 308–318.

[2] Javier Alonso-Mora, Samitha Samaranayake, Alex Wallar, Emilio Frazzoli, and Daniela Rus. 2017. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proc. of the National Academy of Sciences* (2017).

[3] Miguel E. Andrés, Nicolás E. Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. 2013. Geo-Indistinguishability: Differential Privacy for Location-Based Systems. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security* (Berlin, Germany) *(CCS '13)*. 14 pages. https://doi.org/10.1145/2508859.2516735

[4] Timothy Brown. 2016. Matchmaking in Lyft Line — Part 2. eng.lyft.com/matchmaking-in-lyft-line-691a1a32a008.

[5] Konstantinos Chatzikokolakis, Miguel E. Andrés, Nicolás Emilio Bordenabe, and Catuscia Palamidessi. 2013. Broadening the Scope of Differential Privacy Using Metrics. In *Privacy Enhancing Technologies*.

[6] Panayiotis Danassis. 2022. *Scalable Multi-agent Coordination and Resource Sharing*. Ph.D. Dissertation. École Polytechnique Fédérale de Lausanne (EPFL), Lausanne. https://doi.org/10.5075/epfl-thesis-8007

[7] Panayiotis Danassis, Aris Filos-Ratsikas, and Boi Faltings. 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. https://doi.org/10.24963/ijcai.2019/31

[8] Panayiotis Danassis, Marija Sakota, Aris Filos-Ratsikas, and Boi Faltings. 2022. Putting Ridesharing to the Test: Efficient and Scalable Solutions and the Power of Dynamic Vehicle Relocation. *Artificial Intelligence Review* (2022). https://arxiv.org/abs/1912.08066

[9] Panayiotis Danassis, Aleksei Triastcyn, and Boi Faltings. 2022. A Distributed Differentially Private Algorithm for Resource Allocation in Unboundedly Large Settings. *CoRR* abs/2011.07934. arXiv:2011.07934 https://arxiv.org/abs/2011.07934

[10] Panayiotis Danassis, Florian Wiedemair, and Boi Faltings. 2021. Improving Multi-agent Coordination by Learning to Estimate Contention. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. International Joint Conferences on Artificial Intelligence Organization, 125–131. https://doi.org/10.24963/ijcai.2021/18

[11] Cynthia Dwork. 2006. Differential Privacy. In *Automata, Languages and Programming*, Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 1–12.

[12] Cynthia Dwork. 2006. Differential Privacy, In 33rd International Colloquium on Automata, Languages and Programming, part II (ICALP 2006). 4052, 1–12. https://www.microsoft.com/en-us/research/publication/differential-privacy/

[13] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. 2006. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 486–503.

[14] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*.

[15] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science* 9, 3-4 (2014), 211–407.

[16] Ferdinando Fioretto, Chansoo Lee, and Pascal Van Hentenryck. 2018. Constrained-Based Differential Privacy for Mobility Services. In *Proc. of the 17th International Conference on Autonomous Agents and MultiAgent Systems*.

[17] Justin Hsu, Zhiyi Huang, Aaron Roth, Tim Roughgarden, and Zhiwei Steven Wu. 2014. Private Matchings and Allocations. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing* (New York, New York) *(STOC '14)*. Association for Computing Machinery, New York, NY, USA, 21–30. https://doi.org/10.1145/2591796.2591826

[18] Maryam Karimzadehgan, ChengXiang Zhai, and Geneva Belford. 2008. Multi-Aspect Expertise Matching for Review Assignment. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*. https://doi.org/10.1145/1458082.1458230

[19] Harold W. Kuhn. 1955. The Hungarian method for the assignment problem. *Naval Research Logistics* (1955).

[20] László Lovász and Michael D. Plummer. 2009. *Matching theory*. American Mathematical Soc.

[21] Ilya Mironov. 2017. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. IEEE, 263–275.

[22] Amanda Prorok and Vijay Kumar. 2017. Privacy-preserving vehicle assignment for mobility-on-demand systems. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 1869–1876.

[23] Connor Riley, Pascal van Hentenryck, and Enpeng Yuan. 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*.

[24] Alvin E Roth, Tayfun Sönmez, and M Utku Ünver. 2005. Pairwise kidney exchange. *Journal of Economic theory* 125, 2 (2005), 151–188.

[25] Peter Stone, Gal A. Kaminka, Sarit Kraus, and Jeffrey S. Rosenschein. 2010. Ad Hoc Autonomous Agent Teams: Collaboration without Pre-Coordination. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence*.

[26] Hsin-Hao Su. 2015. Algorithms for Fundamental Problems in Computer Networks. (2015).

[27] M. Tang, S. Ow, W. Chen, Y. Cao, K. Lye, and Y. Pan. 2017. The Data and Science behind GrabShare Carpooling. In *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*.

[28] 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.

[29] Wei Tong, Jingyu Hua, and Sheng Zhong. 2017. A Jointly Differentially Private Scheduling Protocol for Ridesharing Services. *IEEE Transactions on Information Forensics and Security* 12, 10 (2017), 2444–2456. https://doi.org/10.1109/TIFS.2017.2707334

[30] Aleksei Triastcyn. 2020. *Data-Aware Privacy-Preserving Machine Learning*. Ph.D. Dissertation. Lausanne. https://doi.org/10.5075/epfl-thesis-7216

[31] Aleksei Triastcyn and Boi Faltings. 2019. Federated Learning with Bayesian Differential Privacy. In *IEEE International Conference on Big Data (Big Data)*. IEEE. https://doi.org/10.1109/BigData47090.2019.9005465

[32] Aleksei Triastcyn and Boi Faltings. 2020. Bayesian Differential Privacy for Machine Learning. In *37th International Conference on Machine Learning*.

[33] Stanley L Warner. 1965. Randomized response: A survey technique for eliminating evasive answer bias. *J. Amer. Statist. Assoc.* 60, 309 (1965), 63–69.