

# Scalable and Jointly Differentially Private Packing

Zhiyi Huang

The University of Hong Kong  
zhiyi@cs.hku.hk

Xue Zhu

The University of Hong Kong  
xuezhu26@hku.hk

---

## Abstract

We introduce an  $(\epsilon, \delta)$ -jointly differentially private algorithm for packing problems. Our algorithm not only achieves the optimal trade-off between the privacy parameter  $\epsilon$  and the minimum supply requirement (up to logarithmic factors), but is also scalable in the sense that the running time is linear in the number of agents  $n$ . Previous algorithms either run in cubic time in  $n$ , or require a minimum supply per resource that is  $\sqrt{n}$  times larger than the best possible.

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Design and analysis of algorithms; Theory of computation  $\rightarrow$  Packing and covering problems

**Keywords and phrases** Joint differential privacy, packing, scalable algorithms

**Digital Object Identifier** 10.4230/LIPIcs.ICALP.2019.73

**Category** Track A: Algorithms, Complexity and Games

**Related Version** A full version of the paper is available at [18], <https://arxiv.org/abs/1905.00767>.

**Funding** *Zhiyi Huang*: This work is supported in by a RGC grant HKU17203717E.

## 1 Introduction

Suppose a trusted principal has  $b$  copies of some privacy-sensitive good, and there are  $n$  agents interested in getting a copy of it. Each agent has some value for receiving a copy of the good. The principal would like to choose a subset of up to  $b$  agents to receive the good so that sum of their values is maximized.<sup>1</sup> However, one of the agents, Alice, gets paranoid that the others may be able to learn a lot of information about her value for the sensitive good. In particular, here is a hypothetical scenario that Alice worries about. Suppose the principal simply allocate to the  $k$  agents with the largest values, with Alice being one of them. Then, all the other  $n - 1$  agents may exchange information and figure out that only  $b - 1$  of them get a copy and, hence, Alice must also get one. Further suppose that the  $b$ -th highest value among them is, say, \$1,000; they would also learn that Alice's value for the sensitive good is at least \$1,000. *Is there an allocation algorithm that addresses Alice's concerns without losing too much in the objective?*

This problem has been studied in a series of works in the last few years [14, 15, 17]. More broadly, let us consider a general packing problem with  $m$  resources and  $n$  agents. Each agent demands a bundle consists of a certain amount of each resource, and has a certain value for getting it. The goal is to pick a subset of the agents such that granting them the corresponding bundles approximately maximizes to sum of the values, subject to the supply constraints of the resources, while protecting the privacy of any individual agent. This line of works focus on a specific notation of privacy called joint differential privacy. In a nutshell,

---

<sup>1</sup> This is also known as social welfare maximization in the literature of mechanism design.



© Zhiyi Huang and Xue Zhu;

licensed under Creative Commons License CC-BY

46th International Colloquium on Automata, Languages, and Programming (ICALP 2019).

Editors: Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi;

Article No. 73; pp. 73:1–73:12



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



it requires that for any individual agent, say, Alice in our example, an adversary shall not be able to learn more than a negligible amount of information about the agent's private information, i.e., her value and demands, from the allocations and prices of the other agents.

How can packing algorithms guarantee joint differential privacy? At a high level, such algorithms leave some amount of supply of each resource unallocated in some meticulous and randomized way, so that even if someone knows the allocations and prices of all other agents as in Alice's hypothetical scenario, he will not be able to learn with certainty whether Alice gets a copy of the item, or her value for it. Further, to ensure that the objective is approximately optimal even with the unallocated supplies, all jointly private algorithms require the supply of each resource to be sufficiently large. Hence, the literature measures how good a jointly private algorithm is by the trade-off between the privacy level, quantified by a parameter  $\epsilon > 0$ , and the minimum supply requirement, subject to getting an additive  $\alpha n$  approximation. To this end, Huang and Zhu [17] show that a supply of  $\frac{\sqrt{m}}{\epsilon\alpha}$  per resource, up to logarithmic factors, is both sufficient and necessary.

Another important consideration is the running time of the algorithms. This is particularly relevant for (jointly) differentially private algorithms, since they generally require the size of the dataset, i.e.,  $n$ , to be sufficiently large to achieve good approximation in the objective. We argue that practical (jointly) differentially private algorithms must be scalable in the sense by Teng [26], i.e., the running time shall be quasi-linear in  $n$  or better. However, the aforementioned algorithm by Huang and Zhu [17], which achieves the optimal  $\tilde{O}(\frac{\sqrt{m}}{\epsilon\alpha})$  supply requirement, is not scalable, as its running time depends cubically in  $n$ . Neither are the earlier algorithms by Hsu et al. [14, 15]. Although Huang and Zhu [17] also propose an alternative algorithm that is scalable, it requires a much larger supply of  $\tilde{O}(\frac{\sqrt{mn}}{\epsilon\alpha})$  per resource. The following question is explicitly left open [17]:

*Are there jointly differentially private algorithm that are scalable and at the same time only require a minimum supply of  $\tilde{O}(\frac{\sqrt{m}}{\epsilon\alpha})$  per resource?*

## 1.1 Our Contributions

We introduce a jointly differentially private packing algorithm that answers the above open question affirmatively. The main theorem of this paper is the following:

► **Theorem 1.** *There is an  $(\epsilon, \delta)$ -jointly differentially private algorithm such that:*

1. *it returns with high probability a feasible packing solution that is optimal up to an  $\alpha n$  additive factor, provided that the supply per resource is at least  $\tilde{O}(\frac{\sqrt{m}}{\alpha\epsilon})$ ;*
2. *it stops in  $O(n)$  time, omitting dependence in other parameters, with high probability.*

The algorithm follows the same high-level framework as the previous ones, which we summarize below. It maintains for each resource a price (per unit of the resource), which can be viewed as a dual variable that Lagrangianizes the corresponding resource constraint of the packing problem. We shall imagine that the prices are posted on a public billboard for everyone to see, including the agents and the adversary. Given the current prices, each agent gets the bundle if and only if her value is higher than the total price of the bundle. The agents' decisions induce a total demand on each resource, which can be viewed as a subgradient for the dual prices w.r.t. some dual objective. The algorithm then increases the prices of the overdemanded resources and decreases those of the underdemanded ones. This process repeats for a certain number of rounds; the final allocation is obtained by averaging over all the rounds. Since the agents' allocation is coordinated only through the prices, it suffices to ensure that the sequence of prices is privacy-preserving. This is formulated as the billboard lemma by Hsu et al. [14].

What are the main differences between our algorithm and the existing ones? The previous non-scalable algorithms are essentially noisy versions of some existing optimization algorithms, including gradient descent [15] and multiplicative weight update [17], run on the dual space with a *fix step size*. They are not scalable because (1) the number of rounds needed by such algorithm generally depends on how large each coordinate of the subgradient could be (a.k.a., the width of the problem), which is roughly  $n$  in our problem, and (2) the time needed to compute the subgradient in each round is linear in  $n$ .

The scalable algorithm by Huang and Zhu [17], on the other hand, is a noisy version of the online multiplicative weight update algorithm (e.g., [2]), which use the demand of a single agent as an estimator of the overall demand in each round, iterating through all agents once in a random order. However, it does not seem plausible to avoid having an extra  $\sqrt{n}$  factor in the minimum supply requirement using this approach, as it is not only jointly differentially private, but also locally private,<sup>2</sup> in the sense that it can be implemented in a way such that the agents add noises themselves so that even the algorithm never accesses any non-private version of the data. The extra  $\sqrt{n}$  factor is ubiquitous in the literature of locally private algorithms [3, 9].

In contrast, our algorithm is a noisy version of the multiplicative weight update algorithm run on the dual space with *different step sizes*, which are optimized according on the scale of the subgradient in each round. Intuitively, it chooses a small step size when the scale of the subgradient is large, to avoid dramatic changes in prices, and a large step size when the scale of the subgradient is small, to ensure a good enough progress. The idea of using different step sizes is widely used in non-private packing algorithms to get width-independent running time (e.g., [20]). Despite being standard in non-private packing, a direct combination of it and how the existing approaches add noises to the subgradients lead to suboptimal minimum supply requirement and/or super-linear running time in  $n$ . Instead, we need to further use different noise scales in different rounds that are tailored to the scales of the subgradients and, by induction, the corresponding step sizes. In a round where the scale of the subgradient is large (respectively, small) and the step size is small (respectively, large), the algorithm adds noises at a larger (respectively, smaller) scale to the subgradient and, in some sense, uses up less (respectively, more) of the privacy budget. Setting noise scales adaptively over time introduces several technical difficulties which we will address in details in the technical sections. To get the results in Theorem 1, the noise scale is inversely proportional to the square root of the step size. Hence, our algorithm is more precisely characterized as a noisy version of the multiplicative weight update algorithm run on the dual space with *different step sizes and noise scales*, both of which are optimized according on the scale of the subgradient in each round. Table 1 provides a brief comparison of our algorithm and the existing ones.

## 1.2 Related Work

The notion of differential privacy is introduced by Dwork et al. [10]. It has evolved through a long line of works to become a standard notion of privacy in theoretical computer science. See Dwork and Roth [12] for a textbook introduction. A particularly related line of works study differentially private algorithms for combinatorial optimization problems and mathematical programs. McSherry and Talwar [22] introduce a generic (yet computationally inefficient) method called the exponential mechanism for privately solving optimization problems whose

---

<sup>2</sup> This is not explicitly stated in Huang and Zhu [17]. Nonetheless, it follows straightforwardly from the definition of the algorithm.

■ **Table 1** A comparison of the algorithm in this paper and those in previous works.  $n$  and  $m$  denote the number of agents and number of resources respectively.  $\epsilon$  and  $\alpha$  quantify the privacy and approximation guarantees respectively.

Reference	Algorithm	Min Supply	Running Time (in $n$ )
Hsu et al. [15]	Dual GD	$\tilde{O}\left(\frac{m}{\alpha\epsilon}\right)$	$O(n^3)$
Huang and Zhu [17]	Dual MWU (fixed step size)	$\tilde{O}\left(\frac{\sqrt{m}}{\alpha\epsilon}\right)$	$O(n^3)$
	Dual Online MWU	$\tilde{O}\left(\frac{\sqrt{mn}}{\alpha\epsilon}\right)$	$O(n)$
<b>This paper</b>	Dual MWU (different step sizes)	$\tilde{O}\left(\frac{\sqrt{m}}{\alpha\epsilon}\right)$	$O(n)$

feasible set of output is independent on the dataset (e.g.,  $k$ -means clustering). It is not applicable to the packing problem considered in this paper, since the set of feasible allocations crucially rely on the dataset. Hsu et al. [16] study what linear programs can be solved in a differentially private manner. Unfortunately, packing linear programs are not among the solvable ones [14].

Subsequently, Kearns et al. [19] introduce a relaxed notion called joint differential privacy. It is still strong enough to provide provable privacy guarantees, but is also flexible enough to allow positive results for problems that cannot be solved under the original notion of differential privacy. This relaxed notion is widely used not only in resource allocation problems [14, 15, 17] such as the packing problem considered in this paper, but also in coordinating large games [24, 6, 23, 19, 7], privacy-preserving learning [5, 25], privacy-preserving surveys [13], privacy-preserving prediction markets [8], etc.

Recently, jointly differentially private resource allocation algorithms find further applications in regularizing strategic behaviors in the problem of learning reserve prices online in strategic environments [21].

## 2 Model

For any positive integer  $\ell$ , let  $[\ell]$  denote the set of integers between 1 and  $\ell$ , i.e.,  $\{1, 2, \dots, \ell\}$ .

### Packing

Consider a packing problem with  $n$  agents and  $m$  resources. Each agent  $i \in [n]$  demands a bundle of resources; let  $a_{ij}$  denote her demand for each resource  $j \in [m]$ . Further, agent  $i$  has value  $v_i$  for getting the bundle, and 0 for not getting it. We assume that  $a_{ij}$ 's and  $v_i$ 's are bounded between 0 and 1, which is standard in the literature of differential privacy. For any agent  $i \in [n]$ , the demands  $a_{ij}$ 's and the value  $v_i$  are her private data. Let  $U = \{(v, a_1, a_2, \dots, a_m) \in [0, 1]^{m+1}\}$  denote the data universe. Let  $\mathcal{D} \in U^n$  denote a dataset of  $n$  agents. For any resource  $j \in [m]$ , let  $b_j$  denote its supply. The goal is then to choose a subset of the agents who get their demanded bundles, such that the sum of the values of the chosen agents is maximized, subject to that the total demand on each resource does not exceed the corresponding supply.

We remark that we can make two simplifying assumptions because the focal point of the jointly differentially private packing problem lies in whether the minimum supply is sufficiently large. First, we may assume without loss of generality (wlog) that the supplies of all resources are equal; otherwise, we may rescale the larger ones down to be equal to the

minimum one. Second, we may focus on fractional solutions wlog; any fractional solution can be converted into an integral one using independent rounding with essentially the same objective and total demands on the resources, since we are in the large supply regime.

Therefore, the problem can be formulated as the following packing linear program:

$$\begin{aligned} & \text{maximize} && \sum_{i \in [n]} v_i x_i \\ & \text{subject to} && \sum_{i \in [n]} a_{ij} x_i \leq b && \forall j \in [m] \\ & && 0 \leq x_i \leq 1 && \forall i \in [n] \end{aligned}$$

## Differential Privacy and Joint Differential Privacy

Next, we formally define differential privacy and joint differential privacy with respect to the packing problem. Two datasets  $\mathcal{D}, \mathcal{D}' \in U^n$  are  $i$ -neighbors if they differ only in the data of the  $i$ -th agent, that is, if  $\mathcal{D}_j = \mathcal{D}'_j$  for all  $j \neq i$ . We simply say that they are neighbors if they are  $i$ -neighbors for some  $i \in [n]$ . Further, let  $X_i = [0, 1]$  denote the set of feasible decision to each agent  $i \in [n]$ . Let  $X = X_1 \times X_2 \times \cdots \times X_n$  denote the set of feasible outcomes, ignoring the supply constraints. The notion of differential privacy by Dwork et al. [10] requires that the allocation of all agents is chosen from similar distributions for any neighboring datasets in the following sense:

► **Definition 2** (Differential Privacy). *A mechanism  $\mathcal{M} : U^n \mapsto X$  is  $(\epsilon, \delta)$ -differentially private if for any neighbors  $\mathcal{D}, \mathcal{D}' \in U^n$ , and any subset of feasible allocations  $S \subseteq X$ :*

$$\Pr[\mathcal{M}(\mathcal{D}) \in S] \leq \exp(\epsilon) \cdot \Pr[\mathcal{M}(\mathcal{D}') \in S] + \delta .$$

The notion of joint differential privacy by Kearns et al. [19], on the other hand, allows the allocation to each agent to depend non-privately on her own data, so long the allocation to the other agents does not. More precisely, the notion is defined as follows:

► **Definition 3** (Joint Differential Privacy). *A mechanism  $\mathcal{M} : U^n \mapsto X$  is  $(\epsilon, \delta)$ -jointly differentially private if for any  $i \in [n]$ , any  $i$ -neighbors  $\mathcal{D}, \mathcal{D}' \in U^n$ , and any subset of feasible allocations to agents other than  $i$ ,  $S_{-i} \subseteq X_{-i}$ :*

$$\Pr[\mathcal{M}(\mathcal{D})_{-i} \in S_{-i}] \leq \exp(\epsilon) \cdot \Pr[\mathcal{M}(\mathcal{D}')_{-i} \in S_{-i}] + \delta .$$

The technical connections between the two notions are best explained by the following billboard lemma by Hsu et al. [14].

► **Lemma 4** (Billboard Lemma). *Suppose  $\mathcal{M} : U^n \mapsto Y$  is  $(\epsilon, \delta)$ -differentially private. Then, for any collection of functions  $f_i : U \times Y \mapsto X_i$ ,  $i \in [n]$ , the mechanism  $\mathcal{M}'$  that allocates to each agent  $i$  with  $f_i(\mathcal{D}_i, \mathcal{M}(\mathcal{D}))$  is  $(\epsilon, \delta)$ -jointly differentially private.*

## 3 Technical Preliminaries

### 3.1 Lagrangian

The Lagrangian of the packing linear program is:

$$\max_{x \in X} \min_{p \in [0, \infty)^m} \sum_{i \in [n]} v_i x_i - \sum_{j \in [m]} p_j \left( \sum_{i \in [n]} a_{ij} x_i - b_j \right) .$$

Let  $L(x, p)$  denote the partial Lagrangian objective, that is:

$$\begin{aligned} L(x, p) &= \sum_{i \in [n]} v_i x_i - \sum_{j \in [m]} \left( \sum_{i \in [n]} a_{ij} x_i - b_j \right) p_j \\ &= \sum_{j \in [m]} b_j p_j + \sum_{i \in [n]} \left( v_i - \sum_{j \in [m]} a_{ij} p_j \right) x_i . \end{aligned}$$

Let  $D(p) = \max_{x \in [X]} L(x, p)$  denote the dual objective. We shall interpret  $p_j$  as the unit price of resource  $j$  for any  $j \in [m]$ . Given any prices  $p$ , an optimal solution  $x^*(p)$  of the optimization problem  $\max_{x \in [X]} L(x, p)$  is defined as, for any  $i \in [n]$ :

$$x_i^*(p) = \begin{cases} 1, & \text{if } v_i - \sum_{j \in [m]} a_{ij} p_j \geq 0; \\ 0, & \text{otherwise.} \end{cases}$$

We have the following envelope theorem (see, e.g., Afriat [1]).

► **Lemma 5** (Envelope Theorem). *Given any prices  $p$ , the derivatives of the Lagrangian objective w.r.t. the prices, i.e.,  $\nabla_p L(x, p)$ , is a sub-gradient of  $D(p)$  when  $x = x^*(p)$ .*

### 3.2 Truncated Laplacian Distributions

The Laplacian distribution, given mean  $\mu \in \mathbf{R}$  and scale parameter  $b > 0$ , is a continuous distribution defined on  $(-\infty, +\infty)$  such that the probability density of any  $x \in \mathbf{R}$  is:

$$\frac{1}{2b} \exp\left(-\frac{|x - \mu|}{b}\right).$$

Let  $\text{Lap}(\mu, b)$  denote this distribution. It has mean  $\mu$  and variance  $2b^2$ .

Further, we will consider the truncated Laplacian distribution with support  $[\mu - 1 + \alpha, \mu + 1 - \alpha]$ , denoted as  $\text{Lap}_{1-\alpha}(\mu, b)$ . The probability density of any  $x$  in the interval is proportional to that of  $\text{Lap}(\mu, b)$ ; the density is 0 if it is outside the interval.  $\text{Lap}_{1-\alpha}(\mu, b)$  also has mean  $\mu$ , and variance  $O(b^2)$ . (See below for a formal statement.)

Given any mean  $\mu \in [-\alpha, \alpha]$  and target standard deviation (up to a constant factor)  $0 < \sigma \leq \alpha$ , let  $\mathcal{N}(\mu, \sigma) = \text{Lap}_{1-\alpha}(\mu, \sigma)$ . We use this notation to emphasize that the family of noise distributions,  $\mathcal{N}(\mu, \sigma)$ 's, can be replaced by distributions other than the truncated Laplacian distributions, as long as they satisfy the following properties. The proofs are deferred to the full version.

► **Lemma 6.** *The noise distributions  $\mathcal{N}(\mu, \sigma)$ 's satisfy that:*

1. *the mean of  $\mathcal{N}(\mu, \sigma)$  is  $\mu$ ;*
2. *the variance of  $\mathcal{N}(\mu, \sigma)$  is at most  $2\sigma^2$ .*

► **Lemma 7.** *Suppose  $-\alpha \leq \mu_1, \mu_2 \leq \alpha$ , and  $0 < \sigma_1, \sigma_2 \leq \alpha$  satisfy that for some  $0 < \eta \leq \alpha$ ,  $\sigma = \max\{\sigma_1, \sigma_2\}$ , and  $\delta > 0$ : (1)  $|\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2}| \leq \frac{\eta}{\alpha\sigma^2}$ ; (2)  $|\mu_1 - \mu_2| \leq \eta$ ; and  $\delta \leq \frac{\eta \ln(2/\delta)}{\sigma}$ . Then, for any  $S \subseteq \mathbf{R}$ , we have:*

$$\Pr_{z \sim \mathcal{N}(\mu_1, \sigma_1)}[z \in S] \leq \exp\left(\frac{4\eta \ln(2/\delta)}{\sigma}\right) \cdot \Pr_{z \sim \mathcal{N}(\mu_2, \sigma_2)}[z \in S] + \delta.$$

## 4 Our Algorithm

The algorithm follows a primal dual approach, running best response on the primal (i.e., allocation  $x$ ) and a noisy version of the multiplicative weight update (MWU) method on the dual (i.e., the prices  $p$ ). Similar approaches are also used in the previous works that study jointly differentially private packing problem (e.g., [15, 17]). The new ingredients of our algorithm are the use of nonuniform step sizes as well as nonuniform noise scales, both of which are meticulously optimized to achieve both scalable running time and the optimal trade-off between privacy and the minimum supply requirement. In contrast, all previous algorithms use a fix step size and a fix noise scale across different rounds.

**Algorithm 1** Private Dual MWU with Optimized Step Sizes and Noise Scales.**Input:**

- Dataset  $\mathcal{D} \in U^n$ , represented by  $a_{ij}$ 's and  $v_i$ 's;
- Supply (per resource)  $b$ .

**Assumptions:**

- $b \geq \tilde{O}\left(\frac{\sqrt{m}}{\alpha\epsilon}\right)$ ;
- We also assume  $n \geq b$  as the problem is trivial otherwise.

**Parameters:**

- Upper bound on dual prices  $p_{\max} = \frac{2n}{b}$ ;
- Initial dual prices  $p^1$  such that  $p_j^1 = \frac{p_{\max}}{m+1}$  for any  $j \in [m+1]$ ;
- Upper bound on the sum of step sizes  $\eta_{\text{sum}} = \frac{\ln(m+1)}{\alpha b}$ .

- 1: **for**  $t = 1, 2, \dots$ , **until**  $\sum_t \eta^t \geq \eta_{\text{sum}}$  (via a private counter) **do**
- 2: Let  $x^t = x^*(p^t)$ , i.e., the best response to  $p^t$  from the primal viewpoint.
- 3: Let  $\nabla_j D(p^t) = b - \sum_{i \in [n]} a_{ij} x_i^t$  for all  $j \in [m]$ ; let  $\nabla_{m+1} D(p^t) = 0$ .
- 4: Let the *step size* and the *noise scale* be:

$$\eta^t = \min \left\{ \frac{\alpha}{b}, \frac{\alpha}{\nabla_1 D(p^t)}, \frac{\alpha}{\nabla_2 D(p^t)}, \dots, \frac{\alpha}{\nabla_m D(p^t)} \right\}, \quad \sigma^t = \frac{\sqrt{m \eta_{\text{sum}} \eta^t \ln(Tm/\delta)}}{\epsilon}.$$

- 5: Draw  $\delta_j^t \sim \mathcal{N}(\mu_j^t, \sigma^t)$  where  $\mu_j^t = \eta^t \nabla_j D(p^t)$  for all  $j \in [m]$ ; let  $\delta_{m+1}^t = 0$ .
- 6: Let  $\hat{p}_j^{t+1} = p_j^t \cdot \exp(-\delta_j^t)$  for all  $j \in [m+1]$ .
- 7: Let  $p^{t+1}$  be such that  $p_j^{t+1} \propto \hat{p}_j^{t+1}$  and  $\sum_{j \in [m+1]} p_j^{t+1} = p_{\max}$ .
- 8: **end for**
- 9: Let  $T$  be the number of iterations in the for loop.

**Output:**  $\bar{x} = \frac{1}{\eta_{\text{sum}}} \sum_{t=1}^T \eta^t x^t$ , where agent  $i$  observes  $\bar{x}_i$ .

See Algorithm 1 for an exposition by pseudocode.

We now describe the algorithm in more details. Let us first explain the basic dynamic of the dual MWU algorithm. For technical reasons, we add a dummy resource with 0 supply, and 0 demands from all agents, and assume that the dual prices sum to a fix and sufficiently large number  $p_{\max}$ . Starting from some initial guess  $p^1$  of the prices, say, with  $p_{\max}$  uniformly distributed among the  $m+1$  coordinates, the MWU algorithm repeatedly calculates:

$$x^t = x^*(p^t) \quad ; \quad p_j^{t+1} \propto p_j^t \cdot \exp(-\eta^t \nabla_j D(p^t)), \quad \forall j \in [m+1]$$

where  $\eta^t > 0$  is the step size of round  $t$ . That is, the allocation  $x^t$  is the best response to  $p^t$ , which induces a subgradient of the dual objective at  $p^t$  by Lemma 5. Then, from  $p^t$  to  $p^{t+1}$ , each coordinate  $j \in [m+1]$  decreases exponentially by an amount proportional to corresponding subgradient, before they are rescaled to sum to  $p_{\max}$ . With appropriate step sizes, standard analysis shows that the weighted average allocation across different rounds, where the weight of each round is its step size, converges to an optimal allocation.<sup>3</sup>

To obtain the desired privacy guarantee, our algorithm updates the prices with zero-mean noises added to the subgradients. Below we discuss the choice of step sizes and noise scales.

**Step Sizes.** Some standard choices of step sizes include uniform step sizes, i.e.,  $\eta^t = \eta$ , which is used in previous works on jointly private packing algorithms [15, 17], decreasing step sizes, e.g.,  $\eta^t = \frac{\eta}{t}$ , and step sizes inversely proportional to the magnitude of the subgradient, which

<sup>3</sup> The prices also converge, although this is not relevant for our analysis.



are what our algorithm uses (see, e.g., Koufogiannakis and Young [20], for an application in the non-private packing problem). Intuitively, our choice of step sizes ensure that at least one coordinate will be updated by an  $\eta$  amount, which in turns lower bounds the amount of progress made in each round. Our algorithm has one caveat, however, as it further caps the step size by an upper bound, which is set to  $\frac{\alpha}{b}$  for technical reasons, so that the noise added in any single round does not affect the result by too much.

**Noise Scales.** Not surprisingly, our first attempt is to add a uniform amount of noise to every round like the previous jointly private algorithms in the literature [14, 15, 17]. This, however, either requires the minimum supply to be much larger than the best possible, or is not scalable. To see why, let us first consider an overly-simplified argument of why uniform noise scales work in the previous algorithms. Suppose the algorithm takes  $T$  rounds in total. By a standard composition theorem of differential privacy (see, e.g., Dwork and Roth [12]), adding noises at a uniform scale  $\tilde{O}(\frac{\sqrt{T}}{\epsilon})$  is sufficient for achieving  $(\epsilon, \delta)$ -joint differential privacy. Then, if the algorithm uses a uniform step size  $\eta$ , by the standard concentration bound, the cumulative noise summing over  $T$  rounds is roughly  $\tilde{O}(\frac{\eta T}{\epsilon})$ . After averaging, this is essentially a fixed amount of noise  $\tilde{O}(\frac{1}{\epsilon})$  independent of  $T$  and  $\eta$ !

With non-uniform step sizes, however, this is no longer true. As a thought experiment, suppose there are  $T' \ll T$  rounds that have large step sizes, say, all equal to  $\eta$ ; the rest of the rounds can be omitted due to negligible step sizes. Note that the uniform noise scale, i.e.,  $\tilde{O}(\frac{\sqrt{T}}{\epsilon})$ , is still determined by the total number of rounds, repeating the above calculation gives that the amount of noise after averaging is  $\tilde{O}(\frac{\sqrt{T'}}{\epsilon\sqrt{T'}}) \gg \tilde{O}(\frac{1}{\epsilon})$ .

The lesson we learned from this though experiment is that the algorithm must choose non-uniform noise scales: smaller noise scales for more important rounds that have larger step sizes; and larger noise scales for less important rounds that have smaller step sizes. More precisely, we optimize the noise scale in each round  $t$  to be inversely proportional to the square root of the step size, i.e.,  $\sqrt{\eta^t}$ ; this is derived from a Cauchy-Schwarz inequality to balance different aspects of the analysis. (Note that  $\delta_j^t$ 's in Algorithm 1 denote the noises added to the subgradients *multiplied by the corresponding step size* and, hence, its scale is proportional to  $\sqrt{\eta^t}$ , rather than inversely proportional to it.)

**Privacy-preserving Stopping Criteria.** Finally, note that the stopping criteria of  $\sum_t \eta^t \geq \eta_{\text{sum}}$  must be implemented approximately in a privacy-preserving manner as well. This can be done by maintaining  $\sum_t \eta^t$  using a standard technique called private counter (e.g., Chan et al. [4], Dwork et al. [11]). For simplicity of exposition, we will omit this standard component and analyze the algorithm assuming the stopping criteria is implemented exactly.

## 5 Utility and Time Complexity

This section sketches the analysis of utility guarantees provided by Algorithm 1 and its time complexity, under the assumption that the supply is sufficiently large, i.e.,  $b \geq \tilde{O}(\frac{\sqrt{m}}{\alpha\epsilon})$ .

### 5.1 No-regret Lemma

We first introduce a technical lemma that will serve as the overarching tool in the analysis of the algorithm's approximation guarantees in terms of the objective and constraint violations, and its running time. It states that if we compare the Lagrangian objective achieved by the sequence of  $x^t$ 's and  $p^t$ 's computed in the algorithm, and what could have been achieved by replacing  $p^t$ 's with an arbitrary but fixed  $p$ , the difference can be bounded. In other words, the dual price sequence has no regret in the terminology of online learning.



► **Lemma 8.** For any  $p$  such that  $\|p\|_1 = p_{max}$ , with high probability, we have:

$$\sum_{t=1}^T \eta^t (L(x^t, p^t) - L(x^t, p)) \leq D_{\text{KL}}(p \| p^1) + \eta_{\text{sum}} \cdot O(\alpha n) .$$

**Proof Sketch of Lemma 8.** We present a proof sketch of a weaker claim that the inequality holds in expectation, which captures the bottleneck of the analysis. Further showing the stronger claim in the lemma takes a (slightly nonstandard) concentration inequality for martingales. See the full version for a complete proof of the lemma.

Readers familiar with this kind of analysis will find it standard and may directly jump to the end to verify that the contribution from the variance term can be bounded given the noise scales chosen in the algorithm. Fix any step  $t$ , we have the followings:

$$\begin{aligned} \mathbf{E}[\eta^t (L(x^t, p^t) - L(x^t, p))] &= \mathbf{E}[\langle \eta^t \nabla D(p^t), p^t - p \rangle] \\ &= \mathbf{E}[\langle \delta^t, p^t - p \rangle] \\ &= \mathbf{E}[\langle \ln \left( \frac{\hat{p}^{t+1}}{p^t} \right), p - p^t \rangle] \\ &= \mathbf{E}[D_{\text{KL}}(p \| p^t) - D_{\text{KL}}(p \| \hat{p}^{t+1}) + D_{\text{KL}}(p^t \| \hat{p}^{t+1})] \\ &\leq \mathbf{E}[D_{\text{KL}}(p \| p^t) - D_{\text{KL}}(p \| p^{t+1}) + D_{\text{KL}}(p^t \| \hat{p}^{t+1}) - D_{\text{KL}}(p^{t+1} \| \hat{p}^{t+1})] \\ &\leq \mathbf{E}[D_{\text{KL}}(p \| p^t) - D_{\text{KL}}(p \| p^{t+1}) + D_{\text{KL}}(p^t \| \hat{p}^{t+1})] . \end{aligned}$$

We abuse notation and let  $\ln \left( \frac{\hat{p}^{t+1}}{p^t} \right)$  denote a vector whose  $j$ -th coordinate is  $\ln \left( \frac{\hat{p}_j^{t+1}}{p_j^t} \right)$  for any  $j \in [m+1]$ , in the 4th line of the above equation. The last two inequalities follow by the generalized Pythagorean theorem and the non-negativity of divergences, respectively.

Summing over  $t \in [T]$ , the first two terms form a telescopic sum; it is bounded by the first term on the right-hand-side of the inequality stated in the lemma. For the last term, we have:

$$\begin{aligned} \mathbf{E}[D_{\text{KL}}(p^t \| \hat{p}^{t+1})] &= \mathbf{E}[\sum_{j=1}^{m+1} (p_j^t \ln \left( \frac{\hat{p}_j^{t+1}}{p_j^t} \right) - p_j^t + \hat{p}_j^{t+1})] \\ &= \mathbf{E}[\sum_{j=1}^{m+1} p_j^t (\delta_j^t - 1 + \exp(-\delta_j^t))] \\ &\leq \mathbf{E}[\sum_{j=1}^{m+1} p_j^t (\delta_j^t)^2] = \mathbf{E}[\sum_{j=1}^{m+1} p_j^t (\mathbf{E}[\delta_j^t]^2 + \mathbf{Var}[\delta_j^t])] . \end{aligned}$$

Note that the step sizes ensure  $\mathbf{E}[\delta_j^t] \leq \alpha$ . The first part on the right-hand-side sums to:

$$\begin{aligned} \mathbf{E}[\sum_{j=1}^{m+1} \sum_{t=1}^T p_j^t \cdot \mathbf{E}[\delta_j^t]^2] &\leq \alpha \mathbf{E}[\sum_{j=1}^{m+1} p_j^t \cdot \sum_{t=1}^T |\mathbf{E}[\delta_j^t]|] \\ &\leq \alpha \mathbf{E}[\sum_{j=1}^{m+1} p_j^t \cdot (2b \sum_{t=1}^T \eta^t - \sum_{t=1}^T \eta^t \mathbf{E}[\delta_j^t])] \\ &= \eta_{\text{sum}} \cdot O(\alpha n) - \alpha \mathbf{E}[\sum_{j=1}^{m+1} p_j^t \sum_{t=1}^T \eta^t \mathbf{E}[\delta_j^t]] \\ &\leq \eta_{\text{sum}} \cdot O(\alpha n) - \alpha \mathbf{E}[\sum_{j=1}^{m+1} p_j^t \sum_{t=1}^T \eta^t \nabla_j D(p^t)] \\ &= \eta_{\text{sum}} \cdot O(\alpha n) - \alpha \mathbf{E}[\sum_{j=1}^{m+1} \eta^t (L(x^t, p^t) - \sum_{i=1}^n v_i x_i^t)] \\ &\leq \eta_{\text{sum}} \cdot O(\alpha n) - \alpha \mathbf{E}[\sum_{j=1}^{m+1} \eta^t L(x^t, p^t)] \leq \eta_{\text{sum}} \cdot O(\alpha n) . \end{aligned}$$

## 73:10 Scalable and Jointly Differentially Private Packing

Finally, the contribution from the variance part sums to:

$$\begin{aligned}
\mathbf{E} \left[ \sum_{j=1}^{m+1} \sum_{t=1}^T p_j^t \cdot \mathbf{Var} [\delta_j^t] \right] &\leq \mathbf{E} \left[ \sum_{j=1}^{m+1} \sum_{t=1}^T p_j^t \cdot 4(\sigma^t)^2 \right] && \text{(Lemma 7)} \\
&\leq \tilde{O} \left( \sum_{t=1}^T \frac{p_{\max} m \eta_{\text{sum}} \eta^t}{\epsilon^2} \right) && \text{(Definition of } \sigma^t \text{'s)} \\
&\leq \tilde{O} \left( \frac{p_{\max} m \eta_{\text{sum}}^2}{\epsilon} \right) \\
&= \eta_{\text{sum}} \cdot \tilde{O} \left( \frac{n}{\alpha \epsilon^2 b^2} \right) \leq \eta_{\text{sum}} \cdot O(\alpha n) . && \text{(Assumption on } b \text{)}
\end{aligned}$$

Putting them together proves the lemma.  $\blacktriangleleft$

By the choice of  $p^1$ ,  $p_{\max}$ ,  $\eta_{\text{sum}}$ , the fact that  $L(x^t, p^t) \geq \text{OPT}$  since  $x^t$ 's are best responses, and the definition of  $\bar{x}$ , we further have it in a simpler form as a corollary.

► **Lemma 9.** *For any  $p$  such that  $\|p\|_1 = p_{\max}$ , with high probability, we have:*

$$\text{OPT} - L(\bar{x}, p) \leq O(\alpha n) .$$

### 5.2 Approximate Optimality

We now argue that the algorithm gets an objective that is optimal up to an  $O(\alpha n)$  additive factor, with the understanding that further improving it to an  $\alpha n$  factor does not affect any of the asymptotic bound. To do so, simply let  $p$  be such that the first  $m$  coordinates are all equal to 0, and the last dummy coordinate equals  $p_{\max}$ . Then, we have that  $L(\bar{x}, p) = \sum_{i=1}^n v_i \bar{x}_i = \text{ALG}$ . The claim then follows from Lemma 9.

### 5.3 Feasibility

Next, we argue that the algorithm provides an allocation  $\bar{x}$  that is approximately feasible in the sense that the total demand for each resource is at most  $(1 + O(\alpha))b$ . To convert it into an exactly feasible solution as stated in Theorem 1, we can simply scale the allocation down by a  $1 + O(\alpha)$  factor, at the cost of further reducing the objective by at most  $O(\alpha n)$ .

Suppose resource  $j^*$  has the largest demand. Let  $s = \sum_{i=1}^n a_{ij^*} \bar{x}_i - b$  the gap between this demand and the supply  $b$ . Let  $p$  be such that all but the  $j^*$ -th coordinate are equal to 0, and the  $j^*$ -th coordinate is equal to  $p_{\max}$ . Then, we have:

$$L(\bar{x}, p) = \text{ALG} - p_{\max} s \leq \left(1 + \frac{s}{b}\right) \text{OPT} - p_{\max} s .$$

By Lemma 9, we get that:

$$\left(p_{\max} - \frac{\text{OPT}}{b}\right) s \leq O(\alpha n) .$$

Putting together with our choice of  $p_{\max} = \frac{2n}{b}$ , and that  $\text{OPT} \leq n$ , we get  $s \leq O(\alpha b)$ .

### 5.4 Time Complexity

We next show that the number of iterations of the for loop is upper bounded by a polynomial of the parameters other than  $n$ . More precisely, we show that  $T \leq O\left(\frac{m \ln(m+1)}{\alpha^2}\right)$ .

To do so, we classify the iterations into  $m + 1$  types, according to how the step size is chosen. If the step size is  $\eta^t = \frac{\alpha}{b}$ , we call it type 0. Otherwise, if the step size is  $\eta^t = \frac{\alpha}{\nabla_j D(p^t)}$  for some  $j \in [m]$ , we call it type  $j$ .

First, the number of type 0 iterations is at most  $\frac{\ln(m+1)}{\alpha^2}$ , before the sum of the step sizes exceeds  $\eta_{\text{sum}} = \frac{\ln(m+1)}{\alpha b}$ .

Next, consider the iterations of type  $j$ , for each resource  $j \in [m]$ . In each iteration  $t$  of type  $j$ , we have that  $\eta^t \nabla_j D(p^t) = \alpha$ . We argue that there cannot be too many such iterations, because  $\sum_{t=1}^T \eta^t \nabla_j D(p^t)$  is upper bounded as a result of the approximate feasibility of  $\bar{x}$ , which follows from Lemma 9 as argued in the Section 5.3.

On the one hand, we have:

$$\sum_{t=1}^T \eta^t \nabla_j D(p^t) = \sum_{t=1}^T \eta^t (\sum_{i=1}^n a_{ij} \bar{x}_i - b) \leq \eta_{\text{sum}} \cdot O(\alpha b) = \tilde{O}(1) .$$

On the other hand, suppose there are  $T_j$  such iterations, we have:

$$\begin{aligned} \sum_{t=1}^T \eta^t \nabla_j D(p^t) &= \sum_{t \in T_j} \alpha + \sum_{t \notin T_j} \eta^t \nabla_j D(p^t) \\ &\geq \sum_{t \in T_j} \alpha + \sum_{t \notin T_j} \eta^t (-b) \geq \alpha |T_j| - 2b \eta_{\text{sum}} = \alpha |T_j| - \frac{2 \ln(m+1)}{\alpha} . \end{aligned}$$

Putting together gives  $|T_j| \leq \frac{3 \ln(m+1)}{\alpha^2}$ .

## 6 Privacy: Proof Sketch

This section sketches the proof that Algorithm 1 is  $(\epsilon, \delta)$ -jointly differentially private. See the full version for a complete argument.

First, consider any fix step  $t$ . Suppose  $\mathcal{D}$  and  $\tilde{\mathcal{D}}$  are two neighboring datasets. Further suppose the random realizations of the first  $t-1$  iterations are such that  $x^1, x^2, \dots, x^{t-1}$  and  $p^1, p^2, \dots, p^{t-1}$  are identical on  $\mathcal{D}$  and  $\tilde{\mathcal{D}}$ . Let us consider the privacy of step  $t$  alone.

In fact, let us focus on a resource  $j \in [m]$ . How does the distribution from which  $\delta_j^t$  is drawn differ on the two datasets? Suppose the step sizes are  $\eta$  and  $\tilde{\eta}$  respectively. Then, by the choice of step sizes in the algorithm and that  $\nabla_j D(p^t)$  differs by at most 1, we have:

$$\left| \frac{1}{\eta} - \frac{1}{\tilde{\eta}} \right| \leq \frac{1}{\alpha} .$$

Suppose the standard deviations are  $\sigma$  and  $\tilde{\sigma}$ , and the means are  $\mu$  and  $\tilde{\mu}$ , respectively on the two datasets. They satisfy that:

$$\begin{aligned} \left| \frac{1}{\sigma^2} - \frac{1}{\tilde{\sigma}^2} \right| &= \frac{\epsilon^2}{m \eta_{\text{sum}} \ln(T/\delta)} \left| \frac{1}{\eta} - \frac{1}{\tilde{\eta}} \right| \leq \frac{\epsilon^2}{\alpha m \eta_{\text{sum}} \ln(Tm/\delta)} = \frac{\eta}{\alpha \sigma^2} , \\ \left| \mu - \tilde{\mu} \right| &\leq \eta \left| \frac{\mu}{\eta} - \frac{\tilde{\mu}}{\tilde{\eta}} \right| + \tilde{\mu} \eta \left| \frac{1}{\eta} - \frac{1}{\tilde{\eta}} \right| \leq \eta + \alpha \eta \alpha^{-1} = 2\eta , \end{aligned}$$

where  $\left| \frac{\mu}{\eta} - \frac{\tilde{\mu}}{\tilde{\eta}} \right| \leq 1$  because it measures the difference in  $\nabla_j D(p^t)$ .

Also it's easy to verify that  $\eta^t \leq \alpha$ . Therefore, by Lemma 7, we have that the update from price  $p_j^t$  to  $\tilde{p}_j^{t+1}$  posted in round  $t$  for resource  $j$  is  $(\epsilon^t, \frac{\delta}{Tm})$ -differentially private for:

$$\epsilon^t = O\left(\frac{\eta^t}{\sigma^t}\right) = \frac{\epsilon \sqrt{\eta^t \ln(Tm/\delta)}}{\sqrt{m \eta_{\text{sum}}}} .$$

Then, intuitively by the composition theorem (see, e.g., Dwork and Roth [12]), the price sequence is  $(\epsilon, \delta)$ -differentially private because:

$$\sum_{t=1}^T \sum_{j=1}^m (\epsilon^t)^2 = O\left(\frac{\epsilon^2 \eta^t \ln^2(Tm/\delta)}{m \eta_{\text{sum}}}\right) = O\left(\frac{\epsilon^2}{\ln(2/\delta)}\right) .$$

However, note that the standard statement of the composition theorem does not directly apply here since the privacy budget  $\epsilon^t$  in each step is chosen adaptively. Fortunately, the underlying argument still goes through. Again, see the full version for details.

Finally, the privacy guarantee of Algorithm 1 follows by the billboard lemma (Lemma 4).

---

**References**

---

- 1 SN Afriat. Theory of maxima and the method of Lagrange. *SIAM Journal on Applied Mathematics*, 20(3):343–357, 1971.
- 2 Shipra Agrawal and Nikhil R Devanur. Fast algorithms for online stochastic convex programming. In *SODA*, pages 1405–1424. SIAM, 2014.
- 3 Raef Bassily and Adam Smith. Local, private, efficient protocols for succinct histograms. In *STOC*, pages 127–135. ACM, 2015.
- 4 TH Hubert Chan, Elaine Shi, and Dawn Song. Private and continual release of statistics. In *ICALP*, pages 405–417. Springer, 2010.
- 5 Rachel Cummings, Stratis Ioannidis, and Katrina Ligett. Truthful linear regression. In *COLT*, pages 448–483, 2015.
- 6 Rachel Cummings, Michael Kearns, Aaron Roth, and Zhiwei Steven Wu. Privacy and truthful equilibrium selection for aggregative games. In *WINE*, pages 286–299. Springer, 2015.
- 7 Rachel Cummings, Katrina Ligett, Jaikumar Radhakrishnan, Aaron Roth, and Zhiwei Steven Wu. Coordination complexity: Small information coordinating large populations. In *ITCS*, pages 281–290. ACM, 2016.
- 8 Rachel Cummings, David M Pennock, and Jennifer Wortman Vaughan. The possibilities and limitations of private prediction markets. In *EC*, pages 143–160. ACM, 2016.
- 9 John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy and statistical minimax rates. In *FOCS*, pages 429–438. IEEE, 2013.
- 10 Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284. Springer, 2006.
- 11 Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N Rothblum. Differential privacy under continual observation. In *STOC*, pages 715–724. ACM, 2010.
- 12 Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.
- 13 Arpita Ghosh, Katrina Ligett, Aaron Roth, and Grant Schoenebeck. Buying private data without verification. In *EC*, pages 931–948. ACM, 2014.
- 14 Justin Hsu, Zhiyi Huang, Aaron Roth, Tim Roughgarden, and Zhiwei Steven Wu. Private matchings and allocations. *SIAM Journal on Computing*, 45(6):1953–1984, 2016.
- 15 Justin Hsu, Zhiyi Huang, Aaron Roth, and Zhiwei Steven Wu. Jointly private convex programming. In *SODA*, pages 580–599. SIAM, 2016.
- 16 Justin Hsu, Aaron Roth, Tim Roughgarden, and Jonathan Ullman. Privately solving linear programs. In *ICALP*, pages 612–624. Springer, 2014.
- 17 Zhiyi Huang and Xue Zhu. Near optimal jointly private packing algorithms via dual multiplicative weight update. In *SODA*, pages 343–357. SIAM, 2018.
- 18 Zhiyi Huang and Xue Zhu. Scalable and Jointly Differentially Private Packing. *CoRR*, abs/1905.00767, 2019. [arXiv:1905.00767](https://arxiv.org/abs/1905.00767).
- 19 Michael Kearns, Mallesh Pai, Aaron Roth, and Jonathan Ullman. Mechanism design in large games: incentives and privacy. In *ITCS*, pages 403–410. ACM, 2014.
- 20 Christos Koufogiannakis and Neal E Young. A nearly linear-time PTAS for explicit fractional packing and covering linear programs. *Algorithmica*, 70(4):648–674, 2014.
- 21 Jinyan Liu, Zhiyi Huang, and Xiangning Wang. Learning Optimal Reserve Price against Non-myopic Bidders. In *NeurIPS*, pages 2042–2052, 2018.
- 22 Frank McSherry and Kunal Talwar. Mechanism Design via Differential Privacy. In *FOCS*, pages 94–103. IEEE Computer Society, 2007.
- 23 Ryan Rogers, Aaron Roth, Jonathan Ullman, and Zhiwei Steven Wu. Inducing approximately optimal flow using truthful mediators. In *EC*, pages 471–488. ACM, 2015.
- 24 Ryan M Rogers and Aaron Roth. Asymptotically truthful equilibrium selection in large congestion games. In *EC*, pages 771–782. ACM, 2014.
- 25 Roshan Shariff and Or Sheffet. Differentially Private Contextual Linear Bandits. In *NeurIPS*, pages 4301–4311, 2018.
- 26 Shang-Hua Teng. Scalable algorithms for data and network analysis. *Foundations and Trends® in Theoretical Computer Science*, 12(1–2):1–274, 2016.