Artificial Intelligence 347 (2025) 104385

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint

Differentially private fair division ☆

Pasin Manurangsi [a], Warut Suksompong [b,*]

[a] Google Research, Thailand
[b] National University of Singapore, Singapore

A B S T R A C T

Fairness and privacy are two important concerns in social decision-making processes such as resource allocation. We initiate the study of privacy in fair division by investigating the fair allocation of indivisible resources using the well-established framework of differential privacy. We present algorithms for approximate envy-freeness and proportionality when two instances are considered to be adjacent if they differ only on the utility of a single agent for a single item. On the other hand, we provide strong negative results for both fairness criteria when the adjacency notion allows the entire utility function of a single agent to change.

1. Introduction

Fairness is a principal concern in numerous social decision-making processes, not least when it comes to allocating scarce resources among interested parties. Whether we divide equipment between healthcare personnel, assign facility time slots to potential users, or distribute office space among working groups in an organization, it is desirable that all parties involved feel fairly treated. While fair division has been studied in economics for several decades [9,37,38,41], the subject has received substantial interest from computer scientists—in particular, artificial intelligence researchers—in recent years, much of which has concentrated on the fair allocation of indivisible resources [2,7,34,39,46,47,49].

The fair division literature typically focuses on satisfying concrete fairness criteria. Two of the most prominent criteria are *envy-freeness* and *proportionality*. In an envy-free allocation, no agent prefers to have another agent's bundle instead of her own. In a proportional allocation, every agent receives value at least $1/n$ of her value for the entire resource, where n denotes the number of agents. As neither of these criteria is always satisfiable,[1] researchers have proposed the relaxations *envy-freeness up to c items (EFc)* and *proportionality up to c items (PROPc)*; here, c is a non-negative integer parameter. Under additive utilities, EFc implies PROPc for every c,[2] and an EF1 allocation (which must also be PROP1) is guaranteed to exist [31].

In addition to fairness, another consideration that has become increasingly important nowadays—as large amounts of data are constantly collected, processed, and analyzed—is privacy. Indeed, an agent participating in a resource allocation procedure may not want other participants to know her preferences if she considers these as sensitive information, for example, if these preferences correspond to the times when she is available to use a facility, or if they represent her valuations for potential team members when distributing employees in an organization. Consequently, a desirable procedure should ensure that an individual participant's preferences cannot be inferred based on the output of the procedure. Achieving privacy alone is trivial, as the procedure can simply ignore the agents' preferences and always output a fixed allocation that it announces publicly in advance. However, it is clear that

☆ A preliminary version of this paper appears in Proceedings of the 37th AAAI Conference on Artificial Intelligence [33].
* Corresponding author.
E-mail address: warut@comp.nus.edu.sg (W. Suksompong).
[1] For example, when there are two agents and only one item which is valuable to both agents.
[2] See the proof of Proposition 2.2(b) in the work of Manurangsi and Suksompong [32].

https://doi.org/10.1016/j.artint.2025.104385
Received 21 April 2023; Received in revised form 10 February 2025; Accepted 8 June 2025
Available online 13 June 2025
0004-3702/© 2025 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Table 1
Overview of our results. We display the smallest c such that there is an ε-DP algorithm that outputs an EFc or PROPc allocation with probability at least $1-\beta$. For simplicity of the bounds, we assume that $m > n^2$, ε is a small constant, and β is sufficiently small (depending only on ε, n and not on m). All upper bounds hold even for connected allocations. Lower bounds for (agent × item)-level DP hold only for connected allocations, but those for agent-level DP hold for arbitrary allocations. The bounds $O(m/n)$ follow trivially from outputting a fixed allocation in which each agent receives $O(m/n)$ items.

		Agent-Level DP	**(Agent × Item)-Level DP**
EF	Upper	$O(m/n)$ (Trivial)	$O(n \log m)$ (Theorem 4.1)
	Lower	$\Omega\left(\sqrt{\frac{m}{n}\log n}\right)$ (Theorem 3.1)	$\Omega(\log m)$ (Theorem 4.9)
PROP	Upper	$O(m/n)$ (Trivial)	$O(\log m)$ (Theorem 4.7)
	Lower	$\Omega\left(\sqrt{\frac{m}{n}}\right)$ (Theorem 3.2)	$\Omega((\log m)/n)$ (Theorem 4.11)

such a procedure can be highly unfair for certain preferences of the agents. Despite its significance, the issue of privacy has been largely unaddressed in the fair division literature as far as we are aware.

In this paper, we investigate the fundamental question of whether fairness and privacy can be attained simultaneously in the allocation of indivisible resources. We use the well-established framework of *differential privacy (DP)*, which has been widely adopted not only in academia but also in industry [3,13,21,24,43] as well as government sectors [1]. Intuitively, the output distribution of a (randomized) DP[3] algorithm should not change by much when a single “entry” of the input is modified. DP provides a privacy protection for individual entries by ensuring that an adversary with access to the output can only gain limited information about each individual entry. At the same time, DP algorithms often still provide useful outputs based on the aggregated information. We outline the tools and concepts from DP used in this work in Section 2.2.[4]

As alluded to above, DP is defined with respect to what one considers to be an entry of the input, or equivalently, in terms of adjacent inputs. We consider two notions of adjacency between fair division instances. For *agent-level adjacency*, two instances are considered to be adjacent if they differ on the utility function of at most one agent. For *(agent × item)-level adjacency*, two instances are adjacent if they differ on at most the utility of a single agent for a single item. We work with the standard definition of *ε-differential privacy (ε-DP)*: for a parameter $\varepsilon \geq 0$, a (randomized) algorithm is said to satisfy ε-DP if the probability that it outputs a certain allocation for an input and the corresponding probability for an adjacent input differ by a factor of at most e^{ε}. Note that, for the same ε, *agent-level DP* offers a stronger privacy protection for the entire utility function of an individual agent, whereas *(agent × item)-level DP* only offers a protection for the utility of a single agent for a single item. Our goal is to devise ε-DP algorithms that output an EFc or PROPc allocation for a small value of c with sufficiently high probability, or to prove that this task is impossible. Denote by n and m the number of agents and items, respectively.

1.1. Our results

We begin in Section 3 by considering agent-level DP. For this demanding benchmark, we establish strong lower bounds with respect to both approximate envy-freeness and proportionality (Theorems 3.1 and 3.2). In both cases, our lower bounds imply that, for fixed n and ε, no ε-DP algorithm can output an EFc or PROPc allocation for $c = o(\sqrt{m})$ with some large constant probability. Our results hold even when the agents have binary additive utilities, and indicate that agent-level DP is too stringent to permit algorithms with significant fairness guarantees.

In Section 4, we turn our attention to (agent × item)-level DP, and deliver encouraging news for this more relaxed notion. In contrast to the previous lower bounds, we present ε-DP algorithms for EFc and PROPc where c only grows logarithmically in m for fixed n and ε (Theorems 4.1 and 4.7).[5] Our EFc algorithm works for arbitrary monotone utility functions, whereas our PROPc algorithm allows (not necessarily binary) additive utilities. Moreover, our algorithms always output allocations that are *connected*.[6] We complement these results by showing a tight lower bound of $\Omega(\log m)$ for connected allocations (Theorems 4.9 and 4.11), even with binary additive utilities.

A summary of our results can be found in Table 1.

1.2. Further related work

The literature of fair division, particularly in the context of indivisible items, has been significantly expanded over the past decade—for an overview of recent progress, we refer readers to the extensive survey by Amanatidis et al. [2]. One of the most

[3] We use the abbreviation DP for both “differential privacy” and “differentially private”.
[4] For in-depth treatments of the subject, we refer to the surveys by Dwork [15] and Dwork and Roth [16].
[5] When n is not fixed, we also provide an improved dependency on n assuming that the utilities are on the “same scale” (Appendix B).
[6] See Section 2 for the definition. Connectivity can be desirable when there is a spatial or temporal order of the items, for instance, when allocating time slots to facility users or offices along a corridor to research groups [4–6,8,45].

stringent fairness notions in this context is *envy-freeness up to any item (EFX)*, which requires any envy from one agent towards another agent to disappear upon removing *any* item from the latter agent's bundle [10]. Except in certain special cases such as the case of three agents [12], it remains open whether an EFX allocation always exists. By contrast, not only is an EF1 allocation guaranteed to exist [31], but such an allocation can also be computed efficiently in several ways. When additional requirements are imposed, even EF1 can become unattainable, thereby necessitating the relaxation to EFc for $c > 1$, where c may depend on m and n; this is the case, for instance, when multiple agents sharing the same set of items must be satisfied simultaneously [32]. As we demonstrate in this paper, a similar need arises under differential privacy constraints.

While our work is the first to address DP in fair division as far as we are aware, DP has been examined in other social choice contexts. Shang et al. [42] investigated DP in rank aggregation and proposed a privacy-preserving algorithm that adds noise to the ranking histogram irrespective of the ranking rule used. Lee [29] presented algorithms that compute approximate winners of voting rules depending on pairwise comparisons—not only do these algorithms require a small number of comparison queries, but they also yield DP guarantees. Li et al. [30] designed DP voting rules based on the Condorcet method, and proved that DP is incompatible with other axioms such as the Condorcet criterion and Pareto efficiency. Hsu et al. [26] investigated DP in the context of matching and showed that, while social welfare maximization cannot be achieved under the DP requirement, the problem becomes feasible if the requirement is relaxed to "joint DP". Kannan et al. [28] examined the setting of barter-exchange economy, where agents are endowed with items and preferences over other agents' items. They demonstrated that no non-trivial approximation of Pareto optimality can be attained along with DP or even joint DP, but is attainable under a further relaxation called "marginal DP".

Our two adjacency notions, agent-level and (agent × item)-level adjacency, are similar to per-person and per-attribute DP defined by Ghazi et al. [23], respectively. As those authors discussed in more detail, (agent × item)-level adjacency is suitable when each agent considers only a few of her item utilities to be sensitive, or alternatively, if the adversary is interested in learning the utilities of only a few items.

2. Preliminaries

In this section, we recapitulate the existing models and results that we will utilize in this paper.

2.1. Fair division

In fair division of indivisible items, there is a set $N = [n]$ of agents and a set $M = [m]$ of items, where $[k]$ denotes the set $\{1, 2, \dots, k\}$ for each positive integer k. The utility function of agent i is given by $u_i : 2^M \to \mathbb{R}_{\geq 0}$. Throughout this work, we assume that utility functions are *monotone*, that is, $u_i(S) \leq u_i(T)$ for any $S \subseteq T \subseteq M$. For a single item $j \in M$, we write $u_i(j)$ instead of $u_i(\{j\})$. We seek to output an *allocation* $A = (A_1, \dots, A_n)$, which is an ordered partition of M into n bundles.

We consider two important fairness notions. Let c be a non-negative integer.

- An allocation A is said to be *envy-free up to c items (EFc)* if, for any $i, i' \in N$, there exists $S \subseteq A_{i'}$ with $|S| \leq c$ such that $u_i(A_i) \geq u_i(A_{i'} \setminus S)$.
- An allocation A is said to be *proportional up to c items (PROPc)* if, for any $i \in N$, there exists $S \subseteq M \setminus A_i$ with $|S| \leq c$ such that $u_i(A_i) \geq u_i(M)/n - u_i(S)$.

We say that an allocation A is *connected* if each A_i corresponds to an interval, i.e., $A_i = \{\ell, \ell+1, \dots, r\}$ for some $\ell, r \in M$; it is possible that A_i is a singleton or empty. Let $\mathcal{P}^{\mathrm{conn}}(m, n)$ denote the set of all connected allocations. We will use the following result on the existence of connected EF2 allocations.[7]

Theorem 2.1 *([6]). For any $m, n \in \mathbb{N}$ and any monotone utility functions, there exists a connected EF2 allocation.*

A utility function u_i is *additive* if $u_i(S) = \sum_{j \in S} u_i(j)$ for all $S \subseteq M$. Furthermore, an additive utility function is said to be *binary* if $u_i(j) \in \{0, 1\}$ for all $j \in M$.

2.2. Differential privacy

Next, we state the setting and important tools of differential privacy. These tools will be useful for our analysis in Section 4.

Let us start by recalling the general definition of DP. Denote by $\mathcal{X}$ the set of all possible inputs to the algorithm.

Definition 2.2 *(Differential Privacy [18])*. Let $\varepsilon \geq 0$ be a non-negative real number. A randomized algorithm $\mathcal{M}$ is said to be *ε-differentially private (ε-DP)* if, for every pair of adjacent inputs X, X', it holds that

$$\Pr[\mathcal{M}(X) = o] \leq e^{\varepsilon} \cdot \Pr[\mathcal{M}(X') = o]$$

for all $o \in \mathrm{range}(\mathcal{M})$.

[7] Recently, Igarashi [27] improved this guarantee to EF1. However, this does not lead to improved guarantees in our setting.

An input in our fair division context consists of the agents' utility functions, while an output is an allocation chosen from a probability distribution over (deterministic) allocations. Different notions of adjacency lead to different levels of privacy protection. We consider two natural notions: agent-level DP and (agent × item)-level DP.

Definition 2.3 *(Agent-Level DP).* Two inputs $(u_i)_{i\in N}$ and $(u'_i)_{i\in N}$ are said to be *agent-level adjacent* if they coincide on all but a single agent, i.e., there exists $i^* \in N$ such that $u_i = u'_i$ for all $i \in N \setminus \{i^*\}$.

An algorithm that is ε-DP against this adjacency notion is said to be *agent-level ε-DP*.

Definition 2.4 *((Agent × Item)-Level DP).* Two inputs $(u_i)_{i\in N}$ and $(u'_i)_{i\in N}$ are said to be *(agent × item)-level adjacent* if they coincide on all but the utility of a single agent for a single item, i.e., there exist $i^* \in N$ and $j^* \in M$ such that

- $u_i = u'_i$ for all $i \in N \setminus \{i^*\}$, and
- $u_{i^*}(S) = u'_{i^*}(S)$ for all $S \subseteq M \setminus \{j^*\}$.

An algorithm that is ε-DP against this adjacency notion is said to be *(agent × item)-level ε-DP*.

Note that agent-level DP provides a stronger privacy protection than (agent × item)-level DP, since every pair of inputs that are (agent × item)-level adjacent is also agent-level adjacent, but not vice versa. Consequently, designing an algorithm for agent-level DP is more difficult than for (agent × item)-level DP. Indeed, we will prove strong lower bounds for agent-level DP and present algorithms for (agent × item)-level DP.

We now outline several tools from the DP literature that will be useful for our proofs concerning (agent × item)-level DP (Section 4).

Basic composition The first tool that we will use is the composition of DP: the result of running multiple DP algorithms remains DP, but with a worse privacy parameter.

Theorem 2.5 *(Basic Composition of DP, e.g., [16]). An algorithm that is a result of running two algorithms (possibly in an adaptive manner) that are ε_1-DP and ε_2-DP, respectively, is $(\varepsilon_1 + \varepsilon_2)$-DP.*

A special case of basic composition (Theorem 2.5) that is often used is the case $\varepsilon_2 = 0$, which is referred to as *post-processing* of DP.

Observation 2.6 *(Post-processing of DP).* An algorithm that runs an ε-DP subroutine and then returns a function of the output of this subroutine is also ε-DP.

Parallel composition While basic composition provides a simple way to account for the privacy budget ε when we run multiple algorithms, it can be improved in certain cases. One such case is when the algorithms are run on "disjoint pieces" of the input. In this case, we do not need to add up the ε's, a fact known as *parallel composition* of DP [35]. Since the statement we use below is slightly different from that in McSherry's work, we provide a full proof for completeness in Appendix A.1.

Theorem 2.7 *([35]). Let $\sim$ be an adjacency notion, and let $\Gamma : \mathcal{X} \to \mathcal{X}^k$ be a function such that, if $X \sim X'$, then there exists $i^* \in [k]$ such that $\Gamma(X)_i = \Gamma(X')_i$ for all $i \in [k] \setminus \{i^*\}$, and $\Gamma(X)_{i^*} \sim \Gamma(X')_{i^*}$. If $\mathcal{M}$ is an ε-DP algorithm with respect to the adjacency notion $\sim$, then the algorithm $\mathcal{M}'$ that outputs $(\mathcal{M}(\Gamma(X)_1), \ldots, \mathcal{M}(\Gamma(X)_k))$ is also ε-DP with respect to $\sim$.*

Group privacy While differential privacy offers protection primarily against the adjacency notion for which it is defined, it also offers protection against more general adjacency notions. Below we state one such protection, which is often referred to as *group differential privacy*.

Let $\sim$ be any adjacency relation. For $k \in \mathbb{N}$, let us define $\sim_k$ as the adjacency relation where $X \sim_k X'$ if and only if there exists a sequence $X_0, \ldots, X_k$ such that $X_0 = X, X_k = X'$, and $X_{i-1} \sim X_i$ for all $i \in [k]$.

Lemma 2.8 *(Group Differential Privacy, e.g., [48]). Let $k \in \mathbb{N}$. If an algorithm is ε-DP with respect to an adjacency notion $\sim$, it is also $(k\varepsilon)$-DP with respect to the adjacency notion $\sim_k$.*

As an immediate consequence of Lemma 2.8, any (agent × item)-level ε-DP algorithm is also agent-level $(m\varepsilon)$-DP. However, the factor $m\varepsilon$ makes the latter guarantee rather weak, especially as m grows.

Sensitivity We next define the *sensitivity* of a function, which will be used multiple times in this work. Note that the definition depends on the adjacency notion, but we do not explicitly include it in the notation for convenience.

Definition 2.9. The *sensitivity* of a function $f : \mathcal{X} \to \mathbb{R}$ (with respect to adjacency notion $\sim$) is defined as $\Delta(f) := \max_{X \sim X'} |f(X) - f(X')|$.

Sensitivity is a key notion in DP, and an important method for exploiting it is the Laplace mechanism.

Definition 2.10. For $b > 0$, the *Laplace distribution* with scale b is the distribution whose probability density function is proportional to $\exp(-|x|/b)$. Given a function $f : \mathcal{X} \to \mathbb{R}$, the *Laplace mechanism* outputs $f(X) + Z$ for each $X \in \mathcal{X}$, where Z is drawn from the Laplace distribution with scale $\Delta(f)/\varepsilon$.

As shown by Dwork et al. [18], the Laplace mechanism satisfies ε-DP. This means that if a function has low sensitivity, then we can estimate it to within a small error (with high probability).

Sparse vector technique We will use the so-called *sparse vector technique (SVT)*. The setting is that there are low-sensitivity functions $f_1, \dots, f_H : \mathcal{X} \to \mathbb{R}$. We want to find the first function f_i whose value is above a target threshold. A straightforward approach would be to add Laplace noise to each $f_i(X)$ and then select the function accordingly; due to the basic composition theorem, this would require us to add noise of scale $O(H/\varepsilon)$ to each function. SVT allows us to reduce the dependency on H to merely $O(\log H)$. The technique was first introduced by Dwork et al. [19], and the convenient version below is due to Dwork and Roth [16, Theorem 3.24].

Theorem 2.11. *There exists a constant $\upsilon > 0$ such that the following holds. Let $f_1, \dots, f_H : \mathcal{X} \to \mathbb{R}$ be functions with $\Delta(f_1), \dots, \Delta(f_H) \le 1$. For any $\varepsilon > 0$, $\beta \in (0,1)$, and $\tau \in \mathbb{R}$, there exists an ε-DP algorithm such that, if $\max_h f_h(X) \ge \tau$, then, with probability at least $1-\beta$, the algorithm outputs $h^* \in [H]$ with the following properties:*

- $f_{h^*}(X) \ge \tau - \upsilon \cdot \log(H/\beta)/\varepsilon$;
- *For all $h' < h^*$, $f_{h'}(X) \le \tau + \upsilon \cdot \log(H/\beta)/\varepsilon$.*

The algorithm runs in time $O(HT)$, where T is the time it takes to evaluate $f_h(X)$ for each $h \in [H]$.

Exponential mechanism We will also use the *exponential mechanism (EM)* of McSherry and Talwar [36]. In its generic form, EM allows us to select a solution from a candidate set $\mathcal{H}$. Specifically, we may define (low-sensitivity) scoring functions $\mathrm{scr}_h : \mathcal{X} \to \mathbb{R}$ for each $h \in \mathcal{H}$. Then, EM outputs a solution that approximately maximizes the score. The precise statement is given below.[8]

Theorem 2.12 *([36]). For any $\varepsilon > 0$ and $\beta \in (0,1)$, a finite set $\mathcal{H}$, and a set of scoring functions $\{\mathrm{scr}_h\}_{h\in\mathcal{H}}$ such that $\Delta(\mathrm{scr}_h) \le 1$ for each $h \in \mathcal{H}$, there is an ε-DP algorithm that, on every input X, outputs h^* such that*

$$\mathrm{scr}_{h^*}(X) \ge \max_{h\in\mathcal{H}} \mathrm{scr}_h(X) - \frac{2\log(|\mathcal{H}|/\beta)}{\varepsilon}$$

with probability at least $1-\beta$. The algorithm runs in time $O(|\mathcal{H}|T)$, where T is the time it takes to evaluate $\mathrm{scr}_h(X)$ for each $h \in \mathcal{H}$.

2.3. Anti-concentration inequalities

Denote by $\mathrm{Ber}(1/2)$ the distribution that is 0 with probability $1/2$, and 1 otherwise. The following type of anti-concentration inequalities is well-known; for completeness, we provide a proof of this version in Appendix A.2.

Lemma 2.13. *Let $k \ge 100$ and let $X_1, \dots, X_k$ be independent random variables drawn from $\mathrm{Ber}(1/2)$. Then,*

$$\Pr\left[\sum_{i=1}^{k} X_i < \frac{k}{2} - 0.1\sqrt{k}\right] \ge \frac{1}{4}.$$

We will also use the following anti-concentration inequality, whose proof can be found in Appendix A.3.

Lemma 2.14. *Let $k \ge 100$, let $X_1, \dots, X_k$ be independent random variables drawn from $\mathrm{Ber}(1/2)$, and let $\gamma \in [2, 2^{k/4}]$. Then,*

$$\Pr\left[\sum_{i=1}^{k} X_i > \frac{k}{2} + 0.1\sqrt{k\log\gamma}\right] \ge \frac{0.1}{\gamma}.$$

3. Agent-level DP

We begin by considering the demanding notion of agent-level DP, and provide strong negative results for this notion.

For EFc, we show a lower bound that,[9] when $m > n\log n$, holds even against $c = \Theta\left(\sqrt{\frac{m}{n}\log n}\right)$.

[8] The formulation here can be derived, e.g., by plugging $t = \log(1/\beta)$ into Corollary 3.12 of Dwork and Roth [16].

[9] Unless specified otherwise, log refers to the natural logarithm.

Theorem 3.1. *There exists a constant $\zeta > 0$ such that, for any $\varepsilon > 0$, there is no agent-level ε-DP algorithm that, for any input binary additive utility functions, outputs an EFc allocation with probability higher than $1 - \frac{e^{-\varepsilon}}{200}$, where $c = \left\lfloor \zeta \sqrt{\frac{m}{n} \cdot \min\left\{\log n, \frac{m}{n}\right\}} \right\rfloor$.*

For proportionality, we prove a slightly weaker bound where $c = \Theta(\sqrt{m/n})$ and the "failure probability" required for the lower bound to apply is also smaller at $O_\varepsilon(1/n)$ (compared to $O_\varepsilon(1)$ for envy-freeness).

Theorem 3.2. *There exists a constant $\zeta > 0$ such that, for any $\varepsilon > 0$, there is no agent-level ε-DP algorithm that, for any input binary additive utility functions, outputs a PROPc allocation with probability higher than $1 - \frac{e^{-\varepsilon}}{8n}$, where $c = \lfloor \zeta \sqrt{m/n} \rfloor$.*

We first prove Theorem 3.2, before proceeding to present the proof of Theorem 3.1, which uses similar arguments but requires a more delicate anti-concentration inequality.

3.1. Proof of Theorem 3.2

We let $\zeta = 0.01$. If $m < 100n$, then $c = 0$ and the theorem holds trivially even without the privacy requirement—for example, if all agents have utility 1 for the same item and utility 0 for the remaining items, then no proportional allocation exists. Hence, we may assume that $m \geq 100n$. Throughout the proof, we consider random utility functions $\boldsymbol{u} = (u_i)_{i \in N}$ where each $u_i(j)$ is an independent $\mathrm{Ber}(1/2)$ random variable. For brevity, we will not repeatedly state this in the calculations below.

We start by proving the following auxiliary lemma that if A_i is small, then, for a random utility $\boldsymbol{u}$ as above, the allocation fails to be PROPc for agent i with a constant probability.

Lemma 3.3. *For $\zeta = 0.01$, let c be as in Theorem 3.2 and A be any allocation, and let $i \in N$ be such that $|A_i| \leq m/n$. Then, we have*

$$\Pr_{\boldsymbol{u}}[A \text{ is not PROP}c \text{ for agent } i] \geq \frac{1}{8}.$$

Proof. Let $c' = 2c$. Recall that A not being PROPc for agent i means that $u_i(A_i) < u_i(M)/n - u_i(S)$ for all $S \subseteq M \setminus A_i$ with $|S| \leq c$. Since u_i is binary additive, $u_i(S) \leq c$. Hence, if $u_i(A_i) < u_i(M)/n - c$, then A is not PROPc for agent i. Therefore, we have

$$\begin{aligned}
\Pr_{\boldsymbol{u}}[A \text{ is not PROP}c \text{ for agent } i] &\geq \Pr_{u_i}\left[u_i(A_i) < \frac{u_i(M)}{n} - c\right] \\
&= \Pr_{u_i}\left[u_i(A_i) < \frac{u_i(A_i)}{n} + \frac{u_i(M \setminus A_i)}{n} - c\right] \\
&= \Pr_{u_i}\left[u_i(A_i) < \frac{u_i(M \setminus A_i)}{n-1} - \frac{n}{n-1} \cdot c\right] \\
&\geq \Pr_{u_i}\left[u_i(A_i) < \frac{m}{2n} - c' \text{ and } u_i(M \setminus A_i) \geq \frac{m(n-1)}{2n}\right] \\
&= \Pr_{u_i}\left[u_i(A_i) < \frac{m}{2n} - c'\right] \cdot \Pr_{u_i}\left[u_i(M \setminus A_i) \geq \frac{m(n-1)}{2n}\right] \\
&\geq \frac{1}{2} \cdot \Pr_{u_i}\left[u_i(A_i) < \frac{m}{2n} - c'\right],
\end{aligned} \tag{1}$$

where the last inequality follows from the fact that $|M \setminus A_i| \geq m(n-1)/n$ and symmetry.

Since $|A_i|$ is an integer, $|A_i| \leq \lfloor m/n \rfloor$. Moreover, since $\lfloor m/n \rfloor \geq 100$ and the function $f(k) = k/2 - 0.1\sqrt{k}$ is increasing in $[1, \infty)$, applying Lemma 2.13 with $k = \lfloor m/n \rfloor$ gives

$$\begin{aligned}
\Pr_{u_i}\left[u_i(A_i) < \frac{m}{2n} - c'\right] &= \Pr_{u_i}\left[u_i(A_i) < \frac{m}{2n} - 2 \cdot \left\lfloor 0.01\sqrt{\frac{m}{n}} \right\rfloor\right] \\
&\geq \Pr_{u_i}\left[u_i(A_i) < \frac{m}{2n} - 0.1\sqrt{\frac{m}{n}}\right] \\
&\geq \Pr_{u_i}\left[u_i(A_i) < \frac{1}{2} \cdot \left\lfloor \frac{m}{n} \right\rfloor - 0.1\sqrt{\left\lfloor \frac{m}{n} \right\rfloor}\right] \geq \frac{1}{4}.
\end{aligned}$$

Plugging this back into (1) yields the desired bound. $\square$

We are now ready to prove Theorem 3.2.

Proof of Theorem 3.2. Let $\zeta = 0.01$ and let $\mathcal{M}$ be any agent-level ε-DP algorithm. Consider the input utility functions $\boldsymbol{u}' = (u'_i)_{i\in N}$ where the utility functions are all-zero, and consider the distribution $\mathcal{M}(\boldsymbol{u}')$. For any allocation A, we have $\Pr_{i\in N}\left[|A_i| \le m/n\right] \ge 1/n$ since at least one bundle A_i must have size at most m/n. This implies that $\Pr_{i\in N,\, A\sim\mathcal{M}(\boldsymbol{u}')}\left[|A_i| \le m/n\right] \ge 1/n$. Thus, there exists $i^* \in N$ such that $\Pr_{A\sim\mathcal{M}(\boldsymbol{u}')}[|A_{i^*}| \le m/n] \ge 1/n$.

Recalling the definition of $\boldsymbol{u}$ from earlier and applying Lemma 3.3, we have[10]

$$\Pr_{\boldsymbol{u},\, A\sim\mathcal{M}(\boldsymbol{u}')}\left[A \text{ is not PROP}c \text{ for agent } i^*\right] \ge \Pr_{A\sim\mathcal{M}(\boldsymbol{u}')}\left[|A_{i^*}| \le \frac{m}{n}\right] \cdot \Pr_{\boldsymbol{u},\, A\sim\mathcal{M}(\boldsymbol{u}')}\left[A \text{ is not PROP}c \text{ for agent } i^* \,\middle|\, |A_{i^*}| \le \frac{m}{n}\right]$$
$$\ge \frac{1}{n}\cdot\frac{1}{8} = \frac{1}{8n}.$$

Hence, there exists $\widehat{u}_{i^*}$ such that

$$\Pr_{A\sim\mathcal{M}(\boldsymbol{u}')}\left[A \text{ is not PROP}c \text{ for agent } i^* \text{ with respect to } \widehat{u}_{i^*}\right] \ge \frac{1}{8n}.$$

Now, let $\widehat{\boldsymbol{u}}$ be the input utility such that $\widehat{u}_i$ is all-zero for each $i \ne i^*$ while $\widehat{u}_{i^*}$ is as above. Notice that $\widehat{\boldsymbol{u}}$ is adjacent to $\boldsymbol{u}'$ under agent-level adjacency. Thus, applying the ε-DP guarantee of $\mathcal{M}$, we get

$$\Pr_{A\sim\mathcal{M}(\widehat{\boldsymbol{u}})}[A \text{ is not PROP}c \text{ for agent } i^* \text{ with respect to } \widehat{u}_{i^*}]$$
$$\ge e^{-\varepsilon}\cdot \Pr_{A\sim\mathcal{M}(\boldsymbol{u}')}[A \text{ is not PROP}c \text{ for agent } i^* \text{ with respect to } \widehat{u}_{i^*}] \ge \frac{e^{-\varepsilon}}{8n}.$$

This completes the proof. □

3.2. Proof of Theorem 3.1

We now proceed to prove Theorem 3.1. As in the proof of Theorem 3.2, we let $\zeta = 0.01$. If $m < 100n$, then $c = 0$ and the theorem holds trivially even without the privacy requirement. Hence, we may assume that $m \ge 100n$. We consider random utility functions $\boldsymbol{u} = (u_i)_{i\in N}$ where each $u_i(j)$ is an indepedent $\mathrm{Ber}(1/2)$ random variable. For brevity, we will not repeatedly state this in the calculations below.

For an allocation A and an agent $i \in N$, we let $\mathrm{rank}(i; A)$ denote the size of the set $\{j \in N \setminus \{i\} \mid |A_j| \ge |A_i|\}$.

Lemma 3.4. *For $\zeta = 0.01$, let c be as in Theorem 3.1 and A be any allocation, and let $\ell \in N$ be such that $\mathrm{rank}(\ell; A) \ge (n-1)/2$. Then, we have*

$$\Pr_{\boldsymbol{u}}[A \textit{ is not EFc for agent } \ell] \ge 0.01.$$

Proof. Without loss of generality, we may relabel the agents so that agent ℓ in the lemma statement becomes agent i and $|A_1| \ge \cdots \ge |A_i| > |A_{i+1}| \ge \cdots \ge |A_n|$. We now have $\mathrm{rank}(i; A) = i - 1 \ge \lceil (n-1)/2 \rceil$.

We have

$$\Pr_{\boldsymbol{u}}[A \text{ is not EF}c \text{ for agent } i] \ge \Pr_{u_i}\left[u_i(A_i) \le \frac{|A_i|}{2}\right]\cdot \Pr_{u_i}\left[A_i \text{ is not EF}c \text{ for agent } i \,\middle|\, u_i(A_i) \le \frac{|A_i|}{2}\right]$$
$$\ge \frac{1}{2}\cdot \Pr_{u_i}\left[A_i \text{ is not EF}c \text{ for agent } i \,\middle|\, u_i(A_i) \le \frac{|A_i|}{2}\right]$$
$$\ge \frac{1}{2}\cdot \Pr_{u_i}\left[\exists j < i, u_i(A_j) > \frac{|A_i|}{2} + c \,\middle|\, u_i(A_i) \le \frac{|A_i|}{2}\right]$$
$$= \frac{1}{2}\cdot\left(1 - \prod_{j\in[i-1]} \Pr_{u_i}\left[u_i(A_j) \le \frac{|A_i|}{2} + c\right]\right),$$

where the second inequality follows from the fact that $u_i(A_i)$ is a sum of $|A_i|$ independent $\mathrm{Ber}(1/2)$ random variables and the last equality follows from independence.

We now consider two cases.

- Case I: $|A_i| < m/(2n)$. Since there are m items and n bundles, we have $|A_1| \ge m/n$. Our assumption that $m \ge 100n$ and our choice of c imply that $|A_i|/2 + c \le (|A_1| - 1)/2$. Hence, we have $\Pr[u_i(A_1) \le |A_i|/2 + c] \le 1/2$. It follows that

[10] Here, PROPc is with respect to the utility functions drawn from $\boldsymbol{u}$.

$$\frac{1}{2}\cdot\left(1-\prod_{j\in[i-1]}\Pr_{\boldsymbol{u}}\left[u_i(A_j)\le\frac{|A_i|}{2}+c\right]\right)\ge\frac{1}{2}\cdot\frac{1}{2}=\frac{1}{4}.$$

- Case II: $|A_i| \ge m/(2n)$. For any $j \in [i-1]$, let $S_j \subseteq A_j$ be any subset of A_j of size $|A_i|$. Applying Lemma 2.14 with $\gamma = \min\{n, 2^{m/(8n)}\}$ and $k = |A_i|$ yields

$$\begin{aligned}\Pr_{\boldsymbol{u}}\left[u_i(A_j)>\frac{|A_i|}{2}+c\right]&\ge\Pr_{\boldsymbol{u}}\left[u_i(S_j)>\frac{|A_i|}{2}+c\right]\\&\ge\Pr_{\boldsymbol{u}}\left[u_i(S_j)>\frac{|A_i|}{2}+0.1\sqrt{|A_i|\log\gamma}\right]\ge\frac{0.1}{\gamma}\ge\frac{0.1}{n},\end{aligned}$$

where the second inequality follows from our choice of parameters c, γ and the fact that $|A_i| \ge m/(2n)$, which imply that $c < 0.1\sqrt{|A_i|\log\gamma}$, and the third inequality from Lemma 2.14. Thus, we have

$$\begin{aligned}\Pr_{\boldsymbol{u}}[A_i \text{ is not EF}c \text{ for agent } i]&\ge\frac{1}{2}\cdot\left(1-\left(1-\frac{0.1}{n}\right)^{i-1}\right)\\&\ge\frac{1}{2}\cdot\left(1-\left(1-\frac{0.1}{n}\right)^{\left\lceil\frac{n-1}{2}\right\rceil}\right)\\&\ge\frac{1}{2}\cdot\left(1-e^{-\frac{0.1}{n}\cdot\left\lceil\frac{n-1}{2}\right\rceil}\right)\ge\frac{1}{2}\cdot\left(1-e^{-\frac{0.1}{3}}\right)\ge 0.01,\end{aligned}$$

where for the third inequality we use the well-known estimate $1 + x \le e^x$, which holds for all real numbers x.

Hence, in both cases, we have

$$\Pr_{\boldsymbol{u}}[A_i \text{ is not EF}c \text{ for agent } i]\ge 0.01,$$

which concludes our proof of the lemma. □

With Lemma 3.4 in hand, we can now prove Theorem 3.1 in a similar manner as we proved Theorem 3.2.

Proof of Theorem 3.1. Let $\zeta = 0.01$ and let $\mathcal{M}$ be any agent-level ε-DP algorithm. Consider the input utility functions $\boldsymbol{u}' = (u'_i)_{i\in N}$ where the utility functions are all-zero, and consider the distribution $\mathcal{M}(\boldsymbol{u}')$. Notice that, for any allocation A, it holds that $\Pr_{i\in N}\left[\text{rank}(i;A)\ge\frac{n-1}{2}\right]\ge 1/2$. As a result, we have that $\Pr_{i\in N,\,A\sim\mathcal{M}(\boldsymbol{u}')}[\text{rank}(i;A)\ge\frac{n-1}{2}]\ge 1/2$. Thus, there exists $i^* \in N$ with the property that $\Pr_{A\sim\mathcal{M}(\boldsymbol{u}')}[\text{rank}(i^*;A)\ge\frac{n-1}{2}]\ge 1/2$.

Applying Lemma 3.4, we get

$$\begin{aligned}\Pr_{\boldsymbol{u},\,A\sim\mathcal{M}(\boldsymbol{u}')}[A \text{ is not EF}c \text{ for agent } i^*]&\ge\Pr_{A\sim\mathcal{M}(\boldsymbol{u}')}\left[\text{rank}(i^*;A)\ge\frac{n-1}{2}\right]\cdot\Pr_{\boldsymbol{u},\,A\sim\mathcal{M}(\boldsymbol{u}')}\left[A \text{ is not EF}c \text{ for } i^*\,\middle|\,\text{rank}(i^*;A)\ge\frac{n-1}{2}\right]\\&\ge\frac{1}{2}\cdot\frac{1}{100}=\frac{1}{200}.\end{aligned}$$

Hence, there exists $\widehat{u}_{i^*}$ such that

$$\Pr_{A\sim\mathcal{M}(\boldsymbol{u}')}\left[A \text{ is not EF}c \text{ for agent } i^* \text{ with respect to } \widehat{u}_{i^*}\right]\ge\frac{1}{200}.$$

Now, let $\widehat{\boldsymbol{u}}$ be the input utility such that $\widehat{u}_i$ is all-zero for each $i \ne i^*$ while $\widehat{u}_{i^*}$ is as above. Notice that $\widehat{\boldsymbol{u}}$ is adjacent to $\boldsymbol{u}'$ under agent-level adjacency. Thus, applying the ε-DP guarantee of $\mathcal{M}$, we get

$$\Pr_{A\sim\mathcal{M}(\widehat{\boldsymbol{u}})}[A \text{ is not EF}c \text{ for agent } i^* \text{ with respect to } \widehat{u}_{i^*}]\ge e^{-\varepsilon}\cdot\Pr_{A\sim\mathcal{M}(\boldsymbol{u}')}[A \text{ is not EF}c \text{ for agent } i^* \text{ with respect to } \widehat{u}_{i^*}]\ge\frac{e^{-\varepsilon}}{200}.$$

This completes the proof. □

4. (Agent × Item)-level DP

In this section, we turn our attention to (agent × item)-level DP, which is a more relaxed notion than agent-level DP. We explore both the possibilities (Section 4.1) and limits (Section 4.2) of private algorithms with respect to this notion.

4.1. Algorithms

In contrast to agent-level DP, we will show that $O_{\varepsilon,n}(\log m)$ upper bounds can be attained in the (agent × item)-level DP setting. Before we do so, let us first explain why straightforward approaches do not work. To this end, assume that utilities are additive and

$u_i(j) \in [0,1]$ for all $i \in N$ and $j \in M$. One may want to estimate $u_i(S)$ for each S using the Laplace mechanism. While the Laplace mechanism guarantees that the estimate has an expected additive error of $O(1/\epsilon)$, this is not useful for obtaining approximate envy-freeness or proportionality guarantees: it is possible that (almost) every good yields utility much less than 1. In this case, additive errors do not translate to any non-trivial EFc or PROPc guarantees. We will therefore develop different—and more robust—comparison methods, which ultimately allow us to overcome the aforementioned issue.

4.1.1. Approximate envy-freeness

Our main algorithmic result for approximate envy-freeness is stated below. We remark that this result holds even for non-additive utility functions.

Theorem 4.1. *For any $\epsilon > 0$ and $\beta \in (0,1]$, there exists an (agent × item)-level ϵ-DP algorithm that, for any input monotone utility functions, outputs a connected EFc allocation with probability at least $1-\beta$, where $c = O\left(1 + \frac{n\log(mn)+\log(1/\beta)}{\epsilon}\right)$. The algorithm runs in time $(mn)^{O(n)}$.*

The high-level idea of our algorithm is to apply the exponential mechanism (Theorem 2.12) to select an allocation among the $\le (mn)^n$ *connected* allocations. This gives us an "error" in the score of $O_\epsilon(\log((mn)^n)) = O_\epsilon(n \cdot \log(mn))$. The question is how to set up the score so that (i) it has low sensitivity and (ii) such an error translates to an approximate envy-freeness guarantee. Our insight is to define the score based on the following modified utility function that "removes" a certain number of most valuable items. Using this utility function, we will then define the scoring function formally in Definition 4.5, and prove that it has low sensitivity in Lemma 4.6.

Definition 4.2. For $\boldsymbol{u} = (u_1, \dots, u_n)$ and $k \in \mathbb{N} \cup \{0\}$, we define $\boldsymbol{u}^{-k} = (u_1^{-k}, \dots, u_n^{-k})$ by

$$u_i^{-k}(S) := \min_{T \subseteq M, |T| \le k} u_i(S \setminus T) \qquad \forall i \in N, S \subseteq M.$$

For example, suppose that $S = \{1,2,3\}$ and u_i is additive with $u_i(1) = 5$, $u_i(2) = 3$, and $u_i(3) = 7$. Then, we have $u_i^{-1}(S) = 8$, $u_i^{-2}(S) = 3$, and $u_i^{-k}(S) = 0$ for $k \ge 3$.

It is clear that $\boldsymbol{u}^{-k}$ inherits the monotonicity of $\boldsymbol{u}$. We next list some simple but useful properties of such utility functions. The first property, whose proof is trivial, relates Definition 4.2 to approximate envy-freeness.

Observation 4.3. Let $k, d \in \mathbb{N} \cup \{0\}$. Any allocation that is EFd with respect to $\boldsymbol{u}^{-k}$ is EF$(d+k)$ with respect to $\boldsymbol{u}$.

The second property is that the d, k values are robust with respect to (agent × item)-level adjacency.

Lemma 4.4. *Let $k \in \mathbb{N}$, $d \in \mathbb{N} \cup \{0\}$, and $\boldsymbol{u}, \tilde{\boldsymbol{u}}$ be any two (agent × item)-level adjacent inputs. If an allocation A is EFd with respect to $\boldsymbol{u}^{-k}$, then it is EF$(d+2)$ with respect to $\tilde{\boldsymbol{u}}^{-(k-1)}$.*

Proof. By definition of (agent × item)-level adjacency, there exist $i^* \in N$ and $j^* \in M$ such that

- $u_i = \tilde{u}_i$ for all $i \in N \setminus \{i^*\}$, and
- $u_{i^*}(S) = \tilde{u}_{i^*}(S)$ for all $S \subseteq M \setminus \{j^*\}$.

Consider any $i, i' \in N$. Since A is EFd with respect to $\boldsymbol{u}^{-k}$, we have

$$u_i^{-k}(A_i) \ge \min_{S \subseteq M, |S| \le d} u_i^{-k}(A_{i'} \setminus S) = \min_{T \subseteq M, |T| \le d+k} u_i(A_{i'} \setminus T). \tag{2}$$

Furthermore, we have

$$\begin{aligned} u_i^{-k}(A_i) &= \min_{T \subseteq M, |T| \le k} u_i(A_i \setminus T) \\ &\le \min_{T \subseteq M, |T| \le k-1} u_i(A_i \setminus (T \cup \{j^*\})) \\ &= \min_{T \subseteq M, |T| \le k-1} \tilde{u}_i(A_i \setminus (T \cup \{j^*\})) \\ &\le \min_{T \subseteq M, |T| \le k-1} \tilde{u}_i(A_i \setminus T) = \tilde{u}_i^{-(k-1)}(A_i). \end{aligned} \tag{3}$$

Moreover,

$$\begin{aligned} \min_{T \subseteq M, |T| \le d+k} u_i(A_{i'} \setminus T) &\ge \min_{T \subseteq M, |T| \le d+k} u_i(A_{i'} \setminus (T \cup \{j^*\})) \\ &= \min_{T \subseteq M, |T| \le d+k} \tilde{u}_i(A_{i'} \setminus (T \cup \{j^*\})) \end{aligned}$$

$$\geq \min_{T\subseteq M,|T|\leq d+k+1} \tilde{u}_i(A_{i'}\setminus T)$$
$$= \min_{S\subseteq M,|S|\leq d+2} \tilde{u}_i^{-(k-1)}(A_{i'}\setminus S). \tag{4}$$

Thus, we can conclude that

$$\tilde{u}_i^{-(k-1)}(A_i) \geq u_i^{-k}(A_i) \geq \min_{T\subseteq M,|T|\leq d+k} u_i(A_{i'}\setminus T) \geq \min_{S\subseteq M,|S|\leq d+2} \tilde{u}_i^{-(k-1)}(A_{i'}\setminus S),$$

where the three inequalities hold due to (3), (2), and (4), respectively. It follows that A is EF($d+2$) with respect to $\tilde{\boldsymbol{u}}^{-(k-1)}$. □

We can now define our scoring function.

Definition 4.5. For an allocation A, $\boldsymbol{u}=(u_1,\dots,u_n)$, and $g\in\mathbb{N}$, define

$$\mathrm{scr}_A^g(\boldsymbol{u}) := -\min\{t\in[g] \mid A \text{ is EF}(2t) \text{ with respect to } \boldsymbol{u}^{-(g-t)}\}.$$

We let $\mathrm{scr}_A^g(\boldsymbol{u})=-g$ if the set above is empty.

For example, suppose that there are two agents and five items, each agent has utility 1 for each item, and $g=3$. Consider the allocation A that gives all five items to agent 2. Since $u_1^{-2}(A_1)=0<3-2=u_1^{-2}(A_2)-2$, whereas $u_1^{-1}(A_1)=0\geq 4-4=u_1^{-1}(A_2)-4$, we have $\mathrm{scr}_A^3(\boldsymbol{u})=-2$.

The following lemma shows that this scoring function has low sensitivity.

Lemma 4.6. *For any allocation A and $g\in\mathbb{N}$, $\Delta(\mathrm{scr}_A^g)\leq 1$.*

Proof. Let $\boldsymbol{u},\tilde{\boldsymbol{u}}$ be any pair of (agent × item)-level adjacent inputs. Assume without loss of generality that $\mathrm{scr}_A^g(\boldsymbol{u})\geq\mathrm{scr}_A^g(\tilde{\boldsymbol{u}})$. Let $t^* := -\mathrm{scr}_A^g(\boldsymbol{u})$.

If $t^*=g$, then $\mathrm{scr}_A^g(\tilde{\boldsymbol{u}})\leq\mathrm{scr}_A^g(\boldsymbol{u})=-g$, so $\mathrm{scr}_A^g(\tilde{\boldsymbol{u}})=-g$.

Otherwise, $t^*\leq g-1$, and A is EF($2t^*$) with respect to $\boldsymbol{u}^{-(g-t^*)}$. Lemma 4.4 ensures that A is EF($2(t^*+1)$) with respect to $\tilde{\boldsymbol{u}}^{-(g-t^*-1)}$. Thus, $\mathrm{scr}_A^g(\tilde{\boldsymbol{u}})\geq -(t^*+1)$.

Since $-t^*=\mathrm{scr}_A^g(\boldsymbol{u})\geq\mathrm{scr}_A^g(\tilde{\boldsymbol{u}})$, we have $|\mathrm{scr}_A^g(\boldsymbol{u})-\mathrm{scr}_A^g(\tilde{\boldsymbol{u}})|\leq 1$ in both cases, completing the proof. □

With all the ingredients ready, we can prove Theorem 4.1 by applying the exponential mechanism with appropriate parameters (see Algorithm 1).

Algorithm 1 (Agent × item)-level ε-DP algorithm for EFc.

Parameter: $\varepsilon>0$ and $\beta\in(0,1]$

1: $g\leftarrow 4\left\lceil 1+\frac{\log((mn)^n/\beta)}{\varepsilon}\right\rceil$

2: **return** the allocation output by the ε-DP exponential mechanism using the scoring function scr_A^g (Definition 4.5) with the candidate set $\mathcal{P}^{\mathrm{conn}}(m,n)$ (defined in Section 2.1)

Proof of Theorem 4.1. Let $g=4\left\lceil 1+\frac{\log((mn)^n/\beta)}{\varepsilon}\right\rceil$. We run the exponential mechanism using the scoring function scr_A^g with the candidate set $\mathcal{P}^{\mathrm{conn}}(m,n)$. By Theorem 2.12 (and Lemma 4.6), this is an ε-DP algorithm that, for each $\boldsymbol{u}$, with probability at least $1-\beta$, outputs an allocation A^* such that

$$\mathrm{scr}_{A^*}^g(\boldsymbol{u}) \geq \max_{A\in\mathcal{P}^{\mathrm{conn}}(m,n)} \mathrm{scr}_A^g(\boldsymbol{u}) - \frac{2\log\left(\frac{|\mathcal{P}^{\mathrm{conn}}(m,n)|}{\beta}\right)}{\varepsilon}.$$

Fix any $\boldsymbol{u}$, and define A^* as above. By Theorem 2.1, there exists a connected allocation A^{EF2} that is EF2 with respect to $\boldsymbol{u}^{-(g-1)}$. This means that $\mathrm{scr}_{A^{\mathrm{EF2}}}^g=-1$. Furthermore, we have[11] $|\mathcal{P}^{\mathrm{conn}}(m,n)|\leq(mn)^n$. Plugging these into the inequality above, we get

$$\mathrm{scr}_{A^*}^g(\boldsymbol{u}) \geq -1-\frac{2\log((mn)^n/\beta)}{\varepsilon} \geq -\frac{g}{2},$$

where the latter inequality follows from our choice of g. Hence, A^* is EF($2z$) with respect to $\boldsymbol{u}^{-(g-z)}$ for some $z\leq g/2$. Invoking Observation 4.3, we find that A^* is EF($g+z$), and therefore EF($3g/2$), with respect to $\boldsymbol{u}$.

[11] Indeed, from the set $M=\{1,2,\dots,m\}$, we can allocate one "block" of items at a time starting from items with lower indices. There are at most m possibilities for the size of the next block, this block can be allocated to one of the (at most n) remaining agents, and we allocate n blocks in total, hence the bound $(mn)^n$.

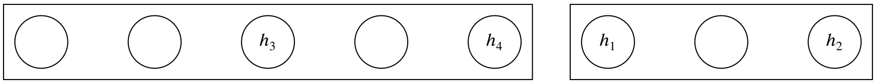

Fig. 1. An example illustrating the top-level recursion of Algorithm 2, with the set of agents $\mathcal{I} = \{1, 2, 3, 4\}$ and the set of items $\{1, 2, \ldots, 8\}$. Since $n^L = 2$, the algorithm recurses on agents 3 and 4 with the set of items $\{1, 2, 3, 4, 5\}$, and on agents 1 and 2 with the set of items $\{6, 7, 8\}$.

Finally, since $|\mathcal{P}^{\mathrm{conn}}(m, n)| \le (mn)^n$ and the score can be computed in polynomial time, the algorithm runs in time $(mn)^{O(n)}$. This concludes our proof. □

If the utilities are additive and on the "same scale", we can achieve an improved dependency on n while keeping the dependency on m polylogarithmic. The details can be found in Appendix B.

4.1.2. Approximate proportionality

Next, we present an improved result for approximate proportionality, where the dependency on n is reduced to $O(\log n)$.

Theorem 4.7. *For any $\varepsilon > 0$ and $\beta \in (0, 1]$, there exists an (agent × item)-level ε-DP algorithm that, for any input additive utility functions, outputs a connected PROPc allocation with probability at least $1 - \beta$, where $c = O\left(\log n + \frac{\log(mn/\beta)}{\varepsilon}\right)$. The algorithm runs in time polynomial in n and m.*

Our algorithm is based on the well-known "moving-knife" procedure from cake cutting [14]. A natural way to implement this idea in our setting is to place the items on a line, put a knife at the left end, and move it rightwards until some agent values the subset of items to the left of the knife at least $1/n$ of her value for the whole set of items. We give this subset to this agent, and proceed similarly with the remaining agents and items. To make this procedure DP, we can replace the check of whether each agent receives sufficiently high value with the SVT algorithm (Theorem 2.11), where the usual utility is modified similarly to Definition 4.5 to achieve low sensitivity. While this approach is feasible, it does not establish the bound we want: since the last agent has to participate in n "rounds" of this protocol, the basic composition theorem (Theorem 2.5) implies that we can only allot a privacy budget of ε/n in each round. This results in a guarantee of the form $c = O(\log m/(\varepsilon/n)) = O((n \log m)/\varepsilon)$, which does not distinctly improve upon the guarantee in Theorem 4.1.

To overcome this issue, notice that instead of targeting a single agent, we can continue moving our knife until at least $n/2$ agents value the subset of items to the left of the knife at least half of the entire set.[12] This allows us to recurse on both sides, thereby reducing the number of rounds to $\log n$. Hence, we may allot a privacy budget of $\varepsilon/\log n$ in each round. Unfortunately, this only results in a bound of the form $c = O(\log m/(\varepsilon/\log n)) = O((\log n \log m)/\varepsilon)$, which is still worse than what we claim in Theorem 4.7.

Our last observation is that we can afford to make more mistakes in earlier rounds: for example, in the first round, we would be fine with making an "error" of roughly $O(n)$ in the knife position because the subsets on both sides will be subdivided to $\Omega(n)$ parts later. As a result, our strategy is to allot less privacy budget in earlier rounds and more in later rounds. By letting the privacy budgets form a geometric sequence, we can achieve our claimed $O(\log(mn)/\varepsilon)$ bound.

Proof of Theorem 4.7. The proof follows the overview outlined above. The statement holds trivially if $\beta = 1$, so assume that $\beta < 1$. For convenience, given any positive integers $\ell \le r$, we write $[\ell, r]$ as a shorthand for $\{\ell, \ell+1, \ldots, r\}$.

Let $\varepsilon_1, \ldots, \varepsilon_{\lceil \log_2 n \rceil}, g_1, \ldots, g_{\lceil \log_2 n \rceil}$ be given by $\varepsilon_b = \frac{\varepsilon}{2 \cdot (1.5^b)}$ and $g_b = 8\lceil \upsilon \cdot \log(mn/\beta)/\varepsilon_b \rceil$, where υ is defined as in Theorem 2.11. Note that g_b is a positive integer divisible by 8 for all b. Our algorithm, DPMovingKnife, is presented as Algorithm 2; an illustration is shown in Fig. 1. The final algorithm is an instantiation of Algorithm 2 with $\mathcal{I} = N$ and $(\ell, r) = (1, m)$.

Since with apply SVT on polynomially many functions $f_h^{i,\ell,r}$ and each $f_h^{i,\ell,r}(\boldsymbol{u})$ can be computed in polynomial time, the overall running time of the algorithm is polynomial.

Before we prove the utility and privacy guarantees of our algorithm, let us prove the following auxiliary lemma, which states that the sensitivity of $f_h^{i,\ell,r}$ is small. The proof is similar to that of Lemma 4.6.

Lemma 4.8. *For all $\ell, r, h \in M$ and $i \in N$, it holds that $\Delta(f_h^{i,\ell,r}) \le 1$.*

Proof. Let $\boldsymbol{u}, \tilde{\boldsymbol{u}}$ be any pair of (agent × item)-level adjacent inputs. Assume without loss of generality that $f_h^{i,\ell,r}(\boldsymbol{u}) \ge f_h^{i,\ell,r}(\tilde{\boldsymbol{u}})$. Let $t^* := f_h^{i,\ell,r}(\boldsymbol{u})$.

If $t^* = 0$, then $f_h^{i,\ell,r}(\tilde{\boldsymbol{u}})$ must also be equal to 0.

Otherwise, $t^* > 0$, and we have

$$\frac{1}{n^L} \cdot u_i^{-(g_b + t^*)}([\ell, h]) \ge \frac{1}{n^R} \cdot u_i^{-(g_b - t^*)}([h+1, r]). \tag{5}$$

[12] Even and Paz [22] used a similar idea in cake cutting.

Algorithm 2 DPMovingKnife.

Input: A set $\mathcal{I} \subseteq N$ of agents, a set $[\ell, r] \subseteq M$ of items
Parameter: $\varepsilon_1, \dots, \varepsilon_{\lceil \log n \rceil} > 0, g_1, \dots, g_{\lceil \log n \rceil} \in \mathbb{N}$
Output: A partial allocation $(A_i)_{i \in \mathcal{I}}$ of $[\ell, r]$

```
if |I| = 1 then
    return (A_i = [ℓ, r]) for i ∈ I
end if
b ← ⌈log_2 |I|⌉
n^R ← ⌊|I|/2⌋
n^L ← |I| − n^R
for each agent i ∈ I do
    for each h ∈ [ℓ, r] do
```

$$T_h^{i,\ell,r}(\boldsymbol{u}) \leftarrow \left\{ t \in [g_b] \;\middle|\; \frac{1}{n^L} \cdot u_i^{-(g_b+t)}([\ell, h]) \geq \frac{1}{n^R} \cdot u_i^{-(g_b-t)}([h+1, r]) \right\}$$

$$f_h^{i,\ell,r}(\boldsymbol{u}) \leftarrow \begin{cases} \max(T_h^{i,\ell,r}(\boldsymbol{u})) & \text{if } T_h^{i,\ell,r}(\boldsymbol{u}) \neq \emptyset; \\ 0 & \text{otherwise} \end{cases}$$

```
    end for
    h_i ← output from the ε_b-DP SVT algorithm (Theorem 2.11) with τ = g_b/2 on f_ℓ^{i,ℓ,r}, ..., f_r^{i,ℓ,r}
end for
h_{z_1} ≤ h_{z_2} ≤ ··· ≤ h_{z_|I|} ← sorted list of h_i's
A^L ← DPMovingKnife({z_1, ..., z_{n^L}}, {ℓ, ..., h_{z_{n^L}}})
A^R ← DPMovingKnife({z_{n^L+1}, ..., z_|I|}, {h_{z_{n^L}} + 1, ..., r})
return the allocation resulting from combining A^L and A^R
```

From this, we can derive

$$\frac{1}{n^L} \cdot \tilde{u}_i^{-(g_b+t^*-1)}([\ell, h]) \geq \frac{1}{n^L} \cdot u_i^{-(g_b+t^*)}([\ell, h]) \geq \frac{1}{n^R} \cdot u_i^{-(g_b-t^*)}([h+1, r]) \geq \frac{1}{n^R} \cdot \tilde{u}_i^{-(g_b-t^*+1)}([h+1, r]),$$

where the first and last inequalities follow from the fact that $\boldsymbol{u}, \tilde{\boldsymbol{u}}$ are (agent $\times$ item)-level adjacent inputs and the middle inequality from (5). Thus, it must be that $f_h^{i,\ell,r}(\tilde{\boldsymbol{u}}) \geq t^* - 1$.

It follows that $|f_h^{i,\ell,r}(u) - f_h^{i,\ell,r}(\tilde{\boldsymbol{u}})| \leq 1$ in both cases, completing the proof. $\square$

We are now ready to establish the privacy and utility guarantees of the algorithm, starting with the former.

Privacy Analysis. We will prove by induction that DPMovingKnife(N, M) is ε-DP. Specifically, we claim that DPMovingKnife$(\mathcal{I}, [\ell, r])$ is $\left(\sum_{b=1}^{\lceil \log_2 |\mathcal{I}| \rceil} \varepsilon_b\right)$-DP for any $\mathcal{I} \subseteq N$ and $[\ell, r] \subseteq M$. We prove this claim by (strong) induction on the size of $\mathcal{I}$. The base case $|\mathcal{I}| = 1$ is trivial.

For $|\mathcal{I}| > 1$, let us divide the algorithm into two stages: (i) computation of $h_{z_1}, \dots, h_{z_{|\mathcal{I}|}}$, and (ii) computation of A^L and A^R given $h_{z_1}, \dots, h_{z_{|\mathcal{I}|}}$. The privacy for each stage can be analyzed as follows:

- **Stage (i).** From Lemma 4.8 and Theorem 2.11, each application of SVT is $(\varepsilon_{\lceil \log_2 |\mathcal{I}| \rceil})$-DP. Moreover, since SVT is applied on each u_i separately, parallel composition (Theorem 2.7) implies[13] that the entire computation of $(h_i)_{i \in \mathcal{I}}$ is also $(\varepsilon_{\lceil \log_2 |\mathcal{I}| \rceil})$-DP. Finally, $h_{z_1}, \dots, h_{z_{|\mathcal{I}|}}$ is simply a post-processing of $(h_i)_{i \in \mathcal{I}}$, so Observation 2.6 ensures that the entire Stage (i) is $(\varepsilon_{\lceil \log_2 |\mathcal{I}| \rceil})$-DP.
- **Stage (ii).** The inductive hypothesis asserts that each recursive call to DPMovingKnife is $\left(\sum_{b=1}^{\lceil \log_2 |\mathcal{I}| \rceil - 1} \varepsilon_b\right)$-DP. Furthermore, since each u_i is used in only one recursive call, we can apply[14] parallel composition (Theorem 2.7), which ensures that the entire Stage (ii) is also $\left(\sum_{b=1}^{\lceil \log_2 |\mathcal{I}| \rceil - 1} \varepsilon_b\right)$-DP.

Therefore, applying basic composition (Theorem 2.5) across the two stages yields that the entire algorithm is $\left(\sum_{b=1}^{\lceil \log_2 |\mathcal{I}| \rceil} \varepsilon_b\right)$-DP. Since $\sum_{b=1}^{\lceil \log_2 n \rceil} \varepsilon_b \leq \sum_{b=1}^{\infty} \varepsilon_b = \varepsilon$, it follows that the entire algorithm when called with $\mathcal{I} = N$ is ε-DP, as desired.

Utility Analysis. We next analyze the utility of the algorithm. To this end, for each agent i, let $\ell_1^i, r_1^i, \dots, \ell_{w_i}^i, r_{w_i}^i$ be the values of ℓ, r with which the algorithm is invoked for a set $\mathcal{I}$ containing i. Note that $\ell_1^i = 1$, $r_1^i = m$, and $w_i \leq \lceil \log_2 n \rceil + 1$. Similarly, let $h_1^i, \dots, h_{w_i-1}^i$ denote the values of h_i output by SVT for agent i in the calls except the last one, and $n_1^i, \dots, n_{w_i}^i$ denote the sizes of $\mathcal{I}$

[13] In particular, we may apply Theorem 2.7 with $\Gamma : \mathcal{X} \to (\mathcal{X})^{|\mathcal{I}|}$ where $\Gamma(\boldsymbol{u})_i = u_i$, and $\mathcal{M}$ being the SVT algorithm.

[14] More specifically, we let $\Gamma : \mathcal{X} \to (\mathcal{X})^2$ where $\Gamma(\boldsymbol{u})_1 = (u_{z_1}, \dots, u_{z_{n^L}})$ and $\Gamma(\boldsymbol{u})_2 = (u_{z_{n^L+1}}, \dots, u_{z_{|\mathcal{I}|}})$, and $\mathcal{M}$ being the DPMovingKnife algorithm.

in the invocations. (Note that we must have $n^i_{w_i} = 1$.) Furthermore, for each $q \in [w_i - 1]$, we let $R^i_q := [\ell^i_q, r^i_q] \setminus [\ell^i_{q+1}, r^i_{q+1}]$ denote the set of items removed in the q-th iteration, and let $u_i^{\text{top-}k}(R^i_q) := \max_{S \subseteq R^i_q, |S| \le k} u_i(S)$ for any non-negative integer k.

Consider any $i \in N$ and $q \in [w_i - 1]$. Let $b_q = \lceil \log_2 n^i_q \rceil$. Notice that $f_r^{i,\ell,r}(\boldsymbol{u}) = g_{b_q} \ge g_{b_q}/2$. Thus, by the guarantee of SVT (Theorem 2.11, taking β/n^2 in place of β), we have that with probability at least $1 - \beta/n^2$, the following holds for each $i \in N$ and $q \in [w_i - 1]$:

$$f^{i,\ell,r}_{h^i_q}(\boldsymbol{u}) \ge \frac{g_{b_q}}{2} - v \cdot \frac{\log(mn^2/\beta)}{\varepsilon_{b_q}} > \frac{g_{b_q}}{2} - 2v \cdot \frac{\log(mn/\beta)}{\varepsilon_{b_q}} \ge \frac{g_{b_q}}{4} \tag{6}$$

and

$$f^{i,\ell,r}_{h'}(\boldsymbol{u}) \le \frac{g_{b_q}}{2} + v \cdot \frac{\log(mn^2/\beta)}{\varepsilon_{b_q}} < \frac{g_{b_q}}{2} + 2v \cdot \frac{\log(mn/\beta)}{\varepsilon_{b_q}} \le \frac{3g_{b_q}}{4} \tag{7}$$

for all $h' < h^i_q$. Therefore, the union bound implies that, with probability at least $1 - \beta$, both (6) and (7) hold for all $i \in N$ and $q \in [w_i - 1]$ simultaneously. We will show that, when this occurs, the output allocation is PROPc for the value of c in the theorem statement.

First, let us fix a pair $i \in N$ and $q \in [w_i - 1]$, and consider the q-th time DPMovingKnife is called with $i \in \mathcal{I}$. We will show that

$$u_i([\ell^i_{q+1}, r^i_{q+1}]) \ge \frac{n^i_{q+1}}{n^i_q} \left(u_i([\ell^i_q, r^i_q]) - u_i^{\text{top-}(2g_{b_q})}(R^i_q) \right). \tag{8}$$

To do so, consider two cases, based on whether i is recursed on the left subinstance[15] (i.e., $i \in \{z_1, \dots, z_{n^L}\}$) or on the right subinstance (i.e., $i \in \{z_{n^L+1}, \dots, z_{n^i_q}\}$).

- Case I: $i \in \{z_1, \dots, z_{n^L}\}$. In this case, we have

$$u_i([\ell^i_{q+1}, r^i_{q+1}]) = u_i([\ell, h_{z_{n^L}}]) \ge u_i([\ell, h^i_q]) \ge \frac{n^L}{n^R} \cdot u_i^{-3g_{b_q}/4}([h^i_q + 1, r]) \ge \frac{n^L}{n^R} \cdot u_i^{-3g_{b_q}/4}([h_{z_{n^L}} + 1, r]),$$

  where for the second inequality we apply (6) on the definition of $f^{i,\ell,r}_{h^i_q}$ with $t \ge g_{b_q}/4$. Therefore, we have

$$\begin{aligned} u_i([\ell^i_{q+1}, r^i_{q+1}]) &\ge \frac{n^L}{|\mathcal{I}|} \cdot u_i([\ell, h_{z_{n^L}}]) + \frac{n^R}{|\mathcal{I}|} \cdot \frac{n^L}{n^R} \cdot u_i^{-3g_{b_q}/4}([h_{z_{n^L}} + 1, r]) \\ &= \frac{n^L}{|\mathcal{I}|} \left(u_i([\ell, h_{z_{n^L}}]) + u_i^{-3g_{b_q}/4}([h_{z_{n^L}} + 1, r]) \right) \\ &= \frac{n^L}{|\mathcal{I}|} \left(u_i([\ell, r]) - u_i^{\text{top-}(3g_{b_q}/4)}([h_{z_{n^L}} + 1, r]) \right) \\ &= \frac{n^i_{q+1}}{n^i_q} \left(u_i([\ell^i_q, r^i_q]) - u_i^{\text{top-}(3g_{b_q}/4)}(R^i_q) \right) \\ &\ge \frac{n^i_{q+1}}{n^i_q} \left(u_i([\ell^i_q, r^i_q]) - u_i^{\text{top-}(2g_{b_q})}(R^i_q) \right). \end{aligned}$$

- Case II: $i \in \{z_{n^L+1}, \dots, z_{n^i_q}\}$. In this case, we have

$$\begin{aligned} u_i([\ell^i_{q+1}, r^i_{q+1}]) &= u_i([h_{z_{n^L}} + 1, r]) \\ &\ge u_i([h^i_q + 1, r]) \\ &\ge u_i^{-1}([h^i_q, r]) \\ &> \frac{n^R}{n^L} \cdot u_i^{-7g_{b_q}/4}([\ell, h^i_q - 1]) \\ &\ge \frac{n^R}{n^L} \cdot u_i^{-(7g_{b_q}/4+1)}([\ell, h^i_q]) \\ &\ge \frac{n^R}{n^L} \cdot u_i^{-(7g_{b_q}/4+1)}([\ell, h_{z_{n^L}}]) \ge \frac{n^R}{n^L} \cdot u_i^{-2g_{b_q}}([\ell, h_{z_{n^L}}]), \end{aligned}$$

  where the third inequality follows from applying (7) with $h' = h^i_q - 1$ and $t = 3g_{b_q}/4$ in the definition of $f^{i,\ell,r}_{h^i_q}$.

[15] Note that all unspecified variables are for the run of the algorithm as specified above.

Therefore, we have

$$
\begin{aligned}
u_i([\ell^i_{q+1}, r^i_{q+1}]) &\geq \frac{n^L}{|\mathcal{I}|} \cdot \frac{n^R}{n^L} \cdot u_i^{-2g_{b_q}}([\ell, h_{z_{n^L}}]) + \frac{n^R}{|\mathcal{I}|} \cdot u_i([h_{z_{n^L}} + 1, r]) \\
&= \frac{n^R}{|\mathcal{I}|} \left(u_i^{-2g_{b_q}}([\ell, h_{z_{n^L}}]) + u_i([h_{z_{n^L}} + 1, r]) \right) \\
&= \frac{n^R}{|\mathcal{I}|} \left(u_i([\ell, r]) - u_i^{\text{top-}(2g_{b_q})}([\ell, h_{z_{n^L}}]) \right) \\
&= \frac{n^i_{q+1}}{n^i_q} \left(u_i([\ell^i_q, r^i_q]) - u_i^{\text{top-}(2g_{b_q})}(R^i_q) \right).
\end{aligned}
$$

Thus, (8) holds in both cases.

Let $A = (A_1, \ldots, A_n)$ be the allocation returned by the algorithm. For each $i \in N$, by repeatedly applying (8), we get

$$
\begin{aligned}
u_i(A_i) &= u_i([\ell^i_{w_i}, r^i_{w_i}]) \\
&\overset{(8)}{\geq} \frac{1}{n^i_{w_i-1}} \left(u_i([\ell^i_{w_i-1}, r^i_{w_i-1}]) - u_i^{\text{top-}(2g_{b_{w_i-1}})}(R^i_{w_i-1}) \right) \\
&\overset{(8)}{\geq} \frac{1}{n^i_{w_i-2}} \left(u_i([\ell^i_{w_i-2}, r^i_{w_i-2}]) - u_i^{\text{top-}(2g_{b_{w_i-2}})}(R^i_{w_i-2}) \right) - \frac{1}{n^i_{w_i-1}} \cdot u_i^{\text{top-}(2g_{b_{w_i-1}})}(R^i_{w_i-1}) \\
&\overset{(8)}{\geq} \cdots \\
&\overset{(8)}{\geq} \frac{1}{n^i_1} \cdot u_i([\ell^i_1, r^i_1]) - \sum_{q \in [w_i-1]} \frac{1}{n^i_q} \cdot u_i^{\text{top-}(2g_{b_q})}(R^i_q) \\
&= \frac{1}{n} \cdot u_i(M) - \sum_{q \in [w_i-1]} \frac{1}{n^i_q} \cdot u_i^{\text{top-}(2g_{b_q})}(R^i_q) \\
&\geq \frac{1}{n} \cdot u_i(M) - \sum_{q \in [w_i-1]} u_i^{\text{top-}(\lceil 2g_{b_q}/n^i_q \rceil)}(R^i_q) \\
&\geq \frac{1}{n} \cdot u_i(M) - u_i^{\text{top-}\left(\sum_{q \in [w_i-1]} \lceil 2g_{b_q}/n^i_q \rceil\right)} \left(\bigcup_{q \in [w_i-1]} R^i_q \right) \\
&= \frac{1}{n} \cdot u_i(M) - u_i^{\text{top-}\left(\sum_{q \in [w_i-1]} \lceil 2g_{b_q}/n^i_q \rceil\right)} \left(M \setminus A_i \right),
\end{aligned}
$$

where the last inequality follows from the fact that $R^i_1, \ldots, R^i_{w_i-1}$ are disjoint.

Now, we can bound the term $\sum_{q \in [w_i-1]} \lceil 2g_{b_q}/n^i_q \rceil$ as follows:

$$
\begin{aligned}
\sum_{q \in [w_i-1]} \left\lceil \frac{2g_{b_q}}{n^i_q} \right\rceil &\leq \sum_{q \in [w_i-1]} \left(\frac{2g_{b_q}}{n^i_q} + 1 \right) \\
&\leq (w_i - 1) + \sum_{q \in [w_i-1]} O\left(\frac{\log(mn/\beta)/\varepsilon_{b_q} + 1}{n^i_q} \right) \\
&\leq O(\log n) + \sum_{q \in [w_i-1]} O\left(\frac{\log(mn/\beta) \cdot 1.5^{b_q}/\varepsilon + 1}{2^{b_q}} \right) \\
&= O(\log n) + \sum_{q \in [w_i-1]} O\left(\log(mn/\beta)/\varepsilon\right) \cdot (0.75)^{b_q} \\
&\leq O(\log n) + O\left(\log(mn/\beta)/\varepsilon\right),
\end{aligned}
$$

where for the second inequality we use the definition $g_{b_q} = 8\lceil \upsilon \cdot \log(mn/\beta)/\varepsilon_{b_q} \rceil$, for the third inequality we use the definitions $b_q = \lceil \log_2 n^i_q \rceil$ and $\varepsilon_{b_q} = \frac{\varepsilon}{2 \cdot (1.5^{b_q})}$, and the last inequality follows from the fact that the b_q's are different for different values of $q \in [w_i - 1]$.

Putting the previous two bounds together, we can conclude that the allocation A is PROPc for $c = O(\log n + \log(mn/\beta)/\varepsilon)$, as desired. □

4.2. Lower bounds

Next, we prove lower bounds for (agent × item)-level DP via the *packing method* [25]. This involves constructing inputs that are close to one another (with respect to the corresponding adjacency notion) such that the acceptable solutions (i.e., EFc or PROPc allocations) are different for different inputs. The DP requirement can then be used to rule out the existence of algorithms with strong utility guarantees. We reiterate that our lower bounds hold only against connected allocations. In our constructions, we design the utility functions so that each input forces us to pick particular positions to cut in order to get an EFc or PROPc allocation. We start with the proof for envy-freeness.

Theorem 4.9. *There exists a constant $\zeta > 0$ such that, for any $\varepsilon \in (0,1]$, there is no ε-DP algorithm that, for any input binary additive utility functions, outputs a connected EFc allocation with probability at least 0.5, where $c = \left\lfloor \zeta \cdot \min\left\{ \frac{\log m}{\varepsilon}, \frac{m}{n}, \sqrt{m} \right\} \right\rfloor$.*

Proof. Let $\zeta = 0.01$, c be as in the theorem statement, and $T = \lfloor m/(4c+4) \rfloor$. We may assume that $c \geq 1$, as otherwise the theorem holds trivially even without the privacy requirement. Consider the following utility functions.

- Let $\boldsymbol{u}' = (u'_1, \dots, u'_n)$ denote the binary additive utility functions defined as follows:
  - u'_1 and u'_2 are all-zero utility functions.
  - For all $i \in \{3, \dots, n\}$ and $j \in M$, let

$$u'_i(j) = \begin{cases} 1 & \text{if } j \geq m - (c+1)(n-2); \\ 0 & \text{otherwise.} \end{cases}$$

- For every $t \in [T]$, let $\boldsymbol{u}^t = (u^t_1, \dots, u^t_n)$ denote the binary additive utility functions defined as follows:
  - u^t_1 and u^t_2 are defined as follows:

$$u^t_1(j) = u^t_2(j) = \begin{cases} 1 & \text{if } \left\lfloor \frac{j-1}{2c+1} \right\rfloor = t-1; \\ 0 & \text{otherwise,} \end{cases}$$

    for all $j \in M$.
  - For all $i \in \{3, \dots, n\}$, u^t_i is exactly the same as u'_i defined earlier.

Suppose for contradiction that there is an ε-DP algorithm $\mathcal{M}$ that, with probability at least 0.5, outputs a connected allocation that is EFc for its input utility functions. For each $t \in [T]$, let $\mathcal{A}^t$ denote the set of allocations that are EFc for $\boldsymbol{u}^t$. The assumption on $\mathcal{M}$ can be written as

$$\Pr[\mathcal{M}(\boldsymbol{u}^t) \in \mathcal{A}^t] \geq 0.5. \tag{9}$$

Let $\sim$ denote the (agent × item)-level adjacency relation. One can check that $\boldsymbol{u}' \sim_{4c+2} \boldsymbol{u}^t$ for all $t \in [T]$.[16] Using this fact together with group DP (Lemma 2.8), we have

$$\Pr[\mathcal{M}(\boldsymbol{u}') \in \mathcal{A}^t] \geq e^{-\varepsilon(4c+2)} \cdot \Pr[\mathcal{M}(\boldsymbol{u}^t) \in \mathcal{A}^t] \geq 0.5 \cdot e^{-\varepsilon(4c+2)}, \tag{10}$$

where the second inequality holds by (9).

Lemma 4.10. *$\mathcal{A}^1, \dots, \mathcal{A}^T$ are disjoint.*

Proof. Suppose for contradiction that $\mathcal{A}^t$ and $\mathcal{A}^{t'}$ overlap for some $t < t'$. Let A be an allocation that is contained in both $\mathcal{A}^t$ and $\mathcal{A}^{t'}$. Observe that A must satisfy the following:

- There must be at least $n-3$ cuts between item $m-(c+1)(n-2)$ and item m. (Otherwise, the allocation cannot be EFc for at least one of agents $3, \dots, n$.)
- There must be a cut between item $(2c+1)(t-1)+1$ and item $(2c+1)t$. (Otherwise, the allocation cannot be EFc for either agent 1 or 2 in $\boldsymbol{u}^t$.)
- There must be a cut between item $(2c+1)(t'-1)+1$ and item $(2c+1)t'$. (Otherwise, the allocation cannot be EFc for either agent 1 or 2 in $\boldsymbol{u}^{t'}$.)

By our parameter selection, we have

$$\max\{(2c+1)t, (2c+1)t'\} \leq (2c+1)T \leq \frac{m}{2} < m - 2cn \leq m - (c+1)(n-2),$$

[16] The notation $\sim_k$ is defined in the paragraph before Lemma 2.8.

where $m - 2cn > m/2$ follows from the assumption that $c \geq 1$. This implies that the intervals of items discussed above are pairwise disjoint, and therefore the cuts must be distinct, i.e., those $n-1$ cuts are all the cuts in A. Now, consider the leftmost interval of A. The cut for this interval must be to the left of item $(2c+1)t$. Suppose that the interval is assigned to agent i. We consider the following two cases.

- Case I: $i \in \{3, \dots, n\}$. In this case, agent i has value 0 for her own bundle. Furthermore, since there are only $n-3$ cuts between item $m-(c+1)(n-2)$ and item m, at least one other bundle has at least $c+1$ items valued 1 by agent i. As such, A cannot be EFc for agent i.
- Case II: $i \in \{1, 2\}$. In $\boldsymbol{u}^{t'}$, agent i values her own bundle zero. Furthermore, since there is just one cut between item $(2c+1)(t'-1)+1$ and item $(2c+1)t'$, at least one other bundle has at least $c+1$ items valued 1 by agent i. As such, the allocation cannot be EFc for agent i.

In both cases, A cannot be EFc for both $\boldsymbol{u}^t$ and $\boldsymbol{u}^{t'}$ simultaneously, a contradiction. This establishes Lemma 4.10. □

Lemma 4.10 implies that

$$1 \geq \sum_{t \in [T]} \Pr[\mathcal{M}(\boldsymbol{u}') \in \mathcal{A}^t] \geq 0.5 \cdot e^{-\varepsilon(4c+2)} \cdot T \geq 0.5 \cdot e^{-6c\varepsilon} \cdot \left\lfloor \frac{m}{8c} \right\rfloor \geq 0.5 \cdot e^{-0.06 \log m} \cdot \lfloor 10\sqrt{m} \rfloor \geq \left(\frac{0.5}{\sqrt{m}}\right) \cdot (5\sqrt{m}) > 1,$$

where the second inequality follows from (10), and the third and fourth inequalities from our choice of parameters c, T together with the assumption that $c \geq 1$. This is a contradiction, which establishes Theorem 4.9. □

For proportionality, we obtain a slightly weaker bound where the $\log m$ term is replaced by $\log(m/n)/n$. The proof is similar to that of Theorem 4.9.

Theorem 4.11. *There exists a constant $\zeta > 0$ such that, for any $\varepsilon \in (0, 1]$, there is no ε-DP algorithm that, for any input binary additive utility functions, outputs a connected PROPc allocation with probability at least 0.5, where $c = \left\lfloor \zeta \cdot \min\left\{ \frac{\log(m/n)}{\varepsilon n}, \frac{m}{n}, \sqrt{\frac{m}{n}} \right\} \right\rfloor$.*

Proof. Let $\zeta = 0.01$, c be as in the theorem statement, and $T = \lfloor m/(2nc+2) \rfloor$. We may assume that $c \geq 1$, as otherwise the theorem holds trivially even without the privacy requirement. Consider the following utility functions.

- Let $\boldsymbol{u}' = (u'_1, \dots, u'_n)$ denote the binary additive utility functions defined as follows:
  - u'_1 and u'_2 are all-zero utility functions.
  - For all $i \in \{3, \dots, n\}$ and $j \in M$, let
$$u'_i(j) = \begin{cases} 1 & \text{if } j \geq m - cn; \\ 0 & \text{otherwise.} \end{cases}$$
- For every $t \in [T]$, let $\boldsymbol{u}^t = (u^t_1, \dots, u^t_n)$ denote the binary additive utility functions defined as follows:
  - u^t_1 and u^t_2 are defined as follows:
$$u^t_1(j) = u^t_2(j) = \begin{cases} 1 & \text{if } \left\lfloor \frac{j-1}{nc+1} \right\rfloor = t-1; \\ 0 & \text{otherwise,} \end{cases}$$
    for all $j \in M$.
  - For all $i \in \{3, \dots, n\}$, u^t_i is exactly the same as u'_i defined earlier.

  Note that every agent values exactly $cn+1$ items, so each agent needs at least one item that she values in order for PROPc to be satisfied.

Suppose for contradiction that there is an ε-DP algorithm $\mathcal{M}$ that, with probability at least 0.5, outputs a connected allocation that is PROPc for its input utility functions. For each $t \in [T]$, let $\mathcal{A}^t$ denote the set of allocations that are PROPc for $\boldsymbol{u}^t$. The assumption on $\mathcal{M}$ can be written as

$$\Pr[\mathcal{M}(\boldsymbol{u}^t) \in \mathcal{A}^t] \geq 0.5. \tag{11}$$

Let $\sim$ denote the (agent × item)-level adjacency relation. One can check that $\boldsymbol{u}' \sim_{2nc+2} \boldsymbol{u}^t$ for all $t \in [T]$. Using this fact together with group DP (Lemma 2.8), we have

$$\Pr[\mathcal{M}(\boldsymbol{u}') \in \mathcal{A}^t] \geq e^{-\varepsilon(2nc+2)} \cdot \Pr[\mathcal{M}(\boldsymbol{u}^t) \in \mathcal{A}^t] \geq 0.5 \cdot e^{-\varepsilon(2nc+2)}, \tag{12}$$

where the second inequality holds by (11).

Lemma 4.12. *$\mathcal{A}^1, \dots, \mathcal{A}^T$ are disjoint.*

Proof. Suppose for contradiction that $\mathcal{A}^t$ and $\mathcal{A}^{t'}$ overlap for some $t < t'$. Let A be an allocation that is contained in both $\mathcal{A}^t$ and $\mathcal{A}^{t'}$. Observe that A must satisfy the following:

- There must be at least $n-3$ cuts between item $m - cn$ and item m. (Otherwise, the allocation cannot be PROPc for at least one of agents $3, \dots, n$.)
- There must be a cut between item $(nc+1)(t-1)+1$ and item $(nc+1)t$. (Otherwise, the allocation cannot be PROPc for either agent 1 or 2 in $\boldsymbol{u}^t$.)
- There must be a cut between item $(nc+1)(t'-1)+1$ and item $(nc+1)t'$. (Otherwise, the allocation cannot be PROPc for either agent 1 or 2 in $\boldsymbol{u}^{t'}$.)

By our parameter selection, we have

$$\max\{(nc+1)t, (nc+1)t'\} \le (nc+1)T \le \frac{m}{2} < m - cn,$$

where the last inequality follows from the assumption that $c \ge 1$. This implies that the intervals of items discussed above are pairwise disjoint, and therefore the cuts must be distinct, i.e., those $n-1$ cuts are all the cuts in A. Now, consider the leftmost interval of A. The cut for this interval must be to the left of item $(nc+1)t$. Suppose that the interval is assigned to agent i. We consider the following two cases.

- Case I: $i \in \{3, \dots, n\}$. In this case, agent i has value 0 for her own bundle, and PROPc is not satisfied.
- Case II: $i \in \{1, 2\}$. In $\boldsymbol{u}^{t'}$, agent i has value 0 for her own bundle, and PROPc is again not satisfied.

In both cases, A cannot be PROPc for both $\boldsymbol{u}^t$ and $\boldsymbol{u}^{t'}$ simultaneously, a contradiction. This establishes Lemma 4.12. □

Lemma 4.12 implies that

$$\begin{aligned} 1 &\ge \sum_{t \in [T]} \Pr[\mathcal{M}(\boldsymbol{u}') \in \mathcal{A}^t] \\ &\ge 0.5 \cdot e^{-\varepsilon(2nc+2)} \cdot T \ge 0.5 \cdot e^{-4nc\varepsilon} \cdot \left\lfloor \frac{m}{4nc} \right\rfloor \ge 0.5 \cdot e^{-0.04\log(m/n)} \cdot \left\lfloor 5\sqrt{\frac{m}{n}} \right\rfloor \ge \left(0.5\left(\frac{m}{n}\right)^{-0.04}\right) \cdot \left(2.5\sqrt{\frac{m}{n}}\right) > 1, \end{aligned}$$

where the second inequality follows from (12), and the third and fourth inequalities from our choice of parameters c, T together with the assumption that $c \ge 1$. This is a contradiction, which establishes Theorem 4.11. □

5. Conclusion and future work

In this paper, we have studied the fundamental task of fair division under differential privacy constraints, and provided algorithms and impossibility results for approximate envy-freeness and proportionality. There are several future directions opened up by our work. First, it would be useful to close the gaps in terms of n—for example, our envy-freeness upper bound for (agent × item)-level DP grows linearly in n (Theorem 4.1) but our lower bound (Theorem 4.9) does not exhibit this behavior. Another perhaps more interesting technical direction is to extend our lower bounds for (agent × item)-level DP to arbitrary (i.e., not necessarily connected) allocations. Specifically, we leave the following intriguing open question: Is there an (agent × item)-level ε-DP algorithm that, with probability at least (say) 0.99, outputs an EFc allocation for $c = O_\varepsilon(1)$ regardless of the values of n and m? Furthermore, it would be interesting to see if one can make the algorithm in Theorem 4.1 run in polynomial time.

While we have considered the original notion of DP proposed by Dwork et al. [18], there are a number of modifications that could be investigated in future work. A commonly studied modification is *approximate DP* (also called *(ε, δ)-DP*), which has an additional parameter $\delta \ge 0$ that specifies the probability with which the condition $\Pr[\mathcal{M}(X) = o] \le e^\varepsilon \cdot \Pr[\mathcal{M}(X') = o]$ is allowed to fail [17]. The notion of DP that we use in this paper corresponds to the case $\delta = 0$ and is often referred to as *pure DP*. Several problems in the literature are known to admit approximate-DP algorithms with better guarantees compared to pure-DP algorithms (see, e.g., the work of Steinke and Ullman [44]). In light of this, a natural direction is to explore whether a similar phenomenon occurs in fair division.

CRediT authorship contribution statement

Pasin Manurangsi: Conceptualization, Formal analysis, Methodology, Writing – original draft, Writing – review & editing. **Warut Suksompong:** Conceptualization, Formal analysis, Methodology, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was partially supported by the Singapore Ministry of Education under grant number MOE-T2EP20221-0001 and by an NUS Start-up Grant. We would like to thank the anonymous reviewers of AAAI 2023 and Artificial Intelligence Journal for their valuable feedback.

Appendix A. Omitted proofs

A.1. Proof of Theorem 2.7

Theorem 2.7 *([35]). Let $\sim$ be an adjacency notion, and let $\Gamma : \mathcal{X} \to \mathcal{X}^k$ be a function such that, if $X \sim X'$, then there exists $i^* \in [k]$ such that $\Gamma(X)_i = \Gamma(X')_i$ for all $i \in [k] \setminus \{i^*\}$, and $\Gamma(X)_{i^*} \sim \Gamma(X')_{i^*}$. If $\mathcal{M}$ is an ε-DP algorithm with respect to the adjacency notion $\sim$, then the algorithm $\mathcal{M}'$ that outputs $(\mathcal{M}(\Gamma(X)_1), \dots, \mathcal{M}(\Gamma(X)_k))$ is also ε-DP with respect to $\sim$.*

Proof. Consider any pair of adjacent inputs $X \sim X'$, and any $o_1, \dots, o_k \in \mathrm{range}(\mathcal{M})$. From the condition on Γ, there exists $i^* \in [k]$ such that $\Gamma(X)_i = \Gamma(X')_i$ for all $i \neq i^*$ and $\Gamma(X)_{i^*} \sim \Gamma(X')_{i^*}$. From this, we have

$$\begin{aligned}
\Pr[\mathcal{M}'(X) = (o_1, \dots, o_k)] &= \Pr[\forall i \in [k],\ \mathcal{M}(\Gamma(X)_i) = o_i] \\
&= \prod_{i \in [k]} \Pr[\mathcal{M}(\Gamma(X)_i) = o_i] \\
&= \left(\prod_{i \in [k] \setminus \{i^*\}} \Pr[\mathcal{M}(\Gamma(X)_i) = o_i] \right) \cdot \Pr[\mathcal{M}(\Gamma(X)_{i^*}) = o_{i^*}] \\
&= \left(\prod_{i \in [k] \setminus \{i^*\}} \Pr[\mathcal{M}(\Gamma(X')_i) = o_i] \right) \cdot \Pr[\mathcal{M}(\Gamma(X)_{i^*}) = o_{i^*}] \\
&\le \left(\prod_{i \in [k] \setminus \{i^*\}} \Pr[\mathcal{M}(\Gamma(X')_i) = o_i] \right) \cdot e^{\varepsilon} \cdot \Pr[\mathcal{M}(\Gamma(X')_{i^*}) = o_{i^*}] \\
&= e^{\varepsilon} \cdot \prod_{i \in [k]} \Pr[\mathcal{M}(\Gamma(X')_i) = o_i] \\
&= e^{\varepsilon} \cdot \Pr[\forall i \in [k],\ \mathcal{M}(\Gamma(X')_i) = o_i] \\
&= e^{\varepsilon} \cdot \Pr[\mathcal{M}'(X') = (o_1, \dots, o_k)],
\end{aligned}$$

where the inequality follows from the ε-DP guarantee of $\mathcal{M}$ and the fact that $\Gamma(X)_{i^*} \sim \Gamma(X')_{i^*}$. Therefore, $\mathcal{M}'$ is ε-DP, as desired. □

A.2. Proof of Lemma 2.13

Lemma 2.13. *Let $k \ge 100$ and let $X_1, \dots, X_k$ be independent random variables drawn from $\mathrm{Ber}(1/2)$. Then,*

$$\Pr\left[\sum_{i=1}^{k} X_i < \frac{k}{2} - 0.1\sqrt{k} \right] \ge \frac{1}{4}.$$

Proof. Let $S = X_1 + \cdots + X_k$. Due to symmetry, we have

$$\Pr\left[S < \frac{k}{2} - 0.1\sqrt{k} \right] = \frac{1}{2}\left(1 - \Pr\left[\left| S - \frac{k}{2} \right| \le 0.1\sqrt{k} \right] \right).$$

Let $b = \lfloor k/2 \rfloor$. We have

$$\Pr\left[\left| S - \frac{k}{2} \right| \le 0.1\sqrt{k} \right] = \sum_{i=\lceil k/2 - 0.1\sqrt{k} \rceil}^{\lfloor k/2 + 0.1\sqrt{k} \rfloor} \binom{k}{i} \cdot \frac{1}{2^k} \le \sum_{i=\lceil k/2 - 0.1\sqrt{k} \rceil}^{\lfloor k/2 + 0.1\sqrt{k} \rfloor} \binom{k}{b} \cdot \frac{1}{2^k} \le \left(0.2\sqrt{k} + 1 \right) \binom{k}{b} \cdot \frac{1}{2^k} \le 0.3\sqrt{k} \cdot \binom{k}{b} \cdot \frac{1}{2^k}.$$

Recall that a more precise version of Stirling's Formula (e.g., [40]) gives

$$\sqrt{2\pi} t^{t+0.5} e^{-t} \le t! \le e^{1/12} \cdot \sqrt{2\pi} t^{t+0.5} e^{-t} \tag{13}$$

for every positive integer t. Now, we may bound $\binom{k}{b} \cdot \frac{1}{2^k}$ as follows:

$$\begin{aligned}
\binom{k}{b} \cdot \frac{1}{2^k} &= \frac{k!}{b!(k-b)!} \cdot \frac{1}{2^k} \\
&\le \left(\frac{e^{1/12}}{\sqrt{2\pi}} \cdot \frac{k^{k+0.5}}{b^{b+0.5}(k-b)^{(k-b)+0.5}} \right) \cdot \frac{1}{2^k} \\
&= \frac{e^{1/12}}{\sqrt{\pi b}} \cdot \left(\frac{k}{2b}\right)^b \cdot \left(\frac{k}{2(k-b)}\right)^{(k-b)+0.5} \\
&\le \frac{e^{1/12}}{\sqrt{\pi b}} \cdot \left(\frac{k}{2b}\right)^b \\
&= \frac{e^{1/12}}{\sqrt{\pi b}} \left(1 + \frac{k-2b}{2b}\right)^b \\
&\le \frac{e^{1/12}}{\sqrt{\pi b}} \left(1 + \frac{1}{2b}\right)^b \le \frac{e^{1/12}}{\sqrt{\pi b}} \cdot e^{(1/2b)\cdot b} = \frac{e^{7/12}}{\sqrt{\pi b}} \le \frac{1.6}{\sqrt{k}},
\end{aligned}$$

where the first inequality follows from (13), the second inequality from $b \le k/2$ which implies $k \le 2(k-b)$, the fourth inequality from the well-known fact that $1 + x \le e^x$ for every real number x, and the last inequality from $b \ge (k-1)/2$ and $k \ge 100$.

Combining the three (in)equalities above, we arrive at

$$\Pr\left[S < \frac{k}{2} - 0.1\sqrt{k}\right] \ge \frac{1}{2}\left(1 - (0.3\sqrt{k})\left(\frac{1.6}{\sqrt{k}}\right)\right) = \frac{1}{2}(1 - 0.48) > \frac{1}{4},$$

completing the proof. □

A.3. Proof of Lemma 2.14

Lemma 2.14. *Let $k \ge 100$, let $X_1, \ldots, X_k$ be independent random variables drawn from* Ber(1/2)*, and let $\gamma \in [2, 2^{k/4}]$. Then,*

$$\Pr\left[\sum_{i=1}^{k} X_i > \frac{k}{2} + 0.1\sqrt{k \log \gamma}\right] \ge \frac{0.1}{\gamma}.$$

Proof. Let $S = X_1 + \cdots + X_k$, $a = \lfloor k/2 + 0.1\sqrt{k\log\gamma} \rfloor + 1$, $b = a + \lfloor 0.2\sqrt{k} \rfloor$, and $r = b - k/2$. Note that by our assumptions, we have

$$(0.4\sqrt{\log\gamma} - 0.2)\sqrt{k} \ge (0.4\sqrt{\log 2} - 0.2)\sqrt{k} \ge 0.1 \cdot 10 = 1.$$

Hence, $0.2\sqrt{k} + 1 \le 0.4\sqrt{k\log\gamma}$, and therefore

$$\begin{aligned}
r &\le 0.1\sqrt{k\log\gamma} + 0.2\sqrt{k} + 1 \le 0.5\sqrt{k\log\gamma} \\
&\le 0.5\sqrt{0.7k\log_2\gamma} \le 0.5\sqrt{0.7k(k/4)} < 0.21k,
\end{aligned} \tag{14}$$

which means that $b < 0.71k$.

We have

$$\begin{aligned}
\Pr\left[S > \frac{k}{2} + 0.1\sqrt{k\log\gamma}\right] &\ge \sum_{i=a}^{b} \Pr[S = i] \\
&= \sum_{i=a}^{b} \binom{k}{i} \cdot \frac{1}{2^k} \\
&\ge (b - a + 1) \cdot \binom{k}{b} \cdot \frac{1}{2^k} \\
&\ge 0.2\sqrt{k}\left(\binom{k}{b} \cdot \frac{1}{2^k}\right) \\
&= 0.2\sqrt{k}\left(\frac{k!}{b!(k-b)!} \cdot \frac{1}{2^k}\right) \\
&\overset{(13)}{\ge} 0.2\sqrt{k}\left(0.3 \cdot \frac{k^{k+0.5}}{b^{b+0.5}(k-b)^{(k-b)+0.5}} \cdot \frac{1}{2^k}\right) \\
&= 0.2\sqrt{k}\left(0.3\sqrt{2} \cdot \frac{1}{b^{0.5}} \cdot \left(\frac{k}{2b}\right)^b \cdot \left(\frac{k}{2(k-b)}\right)^{(k-b)+0.5}\right)
\end{aligned}$$

$$
\begin{aligned}
&= 0.2 \cdot 0.3\sqrt{2} \cdot \sqrt{\frac{k}{b}} \cdot \left(\frac{k}{2b}\right)^b \cdot \left(\frac{k}{2(k-b)}\right)^{(k-b)+0.5} \\
&\geq 0.2 \cdot 0.3\sqrt{2} \cdot \sqrt{\frac{1}{0.71}} \cdot \left(\frac{k}{2b}\right)^b \cdot \left(\frac{k}{2(k-b)}\right)^{(k-b)+0.5} \\
&\geq 0.1 \cdot \left(\frac{k}{2b}\right)^b \cdot \left(\frac{k}{2(k-b)}\right)^{k-b} \\
&= 0.1 \cdot \left(\frac{k}{2b}\right)^{2b-k} \cdot \left(\frac{k^2}{4b(k-b)}\right)^{k-b} \\
&\geq 0.1 \cdot \left(\frac{k}{2b}\right)^{2b-k} \\
&= \frac{0.1}{\left(1+\frac{2r}{k}\right)^{2r}} \\
&\geq \frac{0.1}{\exp(4r^2/k)} \overset{(14)}{\geq} \frac{0.1}{\gamma},
\end{aligned}
$$

where for the third-to-last inequality we use the fact that $4b(k-b) \leq k^2$, which follows from the inequality of arithmetic and geometric means, and for the penultimate inequality we use the well-known fact that $1 + x \leq \exp(x)$ for every real number x. This completes the proof. □

Appendix B. Approximate envy-freeness for same-scale utilities

For $\gamma \in (0, 1]$, we say that an additive utility function u is *γ-same-scale* if $u_i(j) \in \{0\} \cup [\gamma, 1]$ for all $i \in N$ and $j \in M$. We give an improved dependency on n for envy-freeness approximation with such utilities compared to Theorem 4.1, while keeping the dependency on m polylogarithmic. To this end, we need an additional tool.

Range query In the range query problem, we are given a vector $v \in \mathbb{R}^m$ and the goal is to output $(e_{\ell,r})_{1\leq \ell\leq r\leq m}$, where $e_{\ell,r}$ is an estimate for $\sum_{j=\ell}^r v_j$. Two vectors v, v' are considered adjacent if $\|v - v'\|_1 \leq 1$. The error of the output is defined as $\max_{1\leq \ell \leq r\leq m} \left|e_{\ell,r} - \sum_{j=\ell}^r v_j\right|$. Dwork et al. [20] and Chan et al. [11] gave an algorithm for answering range queries with DP with the following guarantee.[17]

Theorem B.1 *([11]). There exists a constant $\zeta > 0$ such that the following holds: For any $\varepsilon > 0$, $\beta \in (0, 1]$, and $m \in \mathbb{N}$, there exists an ε-DP algorithm for the range query problem whose output has error at most $\zeta \cdot \log^{2.5}(m/\beta)/\varepsilon$ with probability $1 - \beta$. The algorithm runs in time $O(m)$.*

Our result follows rather directly from the DP range query algorithm.

Theorem B.2. *For any $\varepsilon > 0$ and $\gamma, \beta \in (0, 1]$, there exists an (agent × item)-level ε-DP algorithm that, for any input γ-same-scale additive utility functions, outputs a connected EFc allocation with probability $1 - \beta$ for $c = O((\log^{2.5}(mn/\beta)/\varepsilon + 1)/\gamma)$. The algorithm runs in time $(mn)^{O(n)}$.*

Proof. Let $c = 10 \cdot (\zeta \cdot \log^{2.5}(mn/\beta)/\varepsilon + 1)/\gamma$, where ζ is as in Theorem B.1. The algorithm works as follows:

- For each agent $i \in N$, run the ε-DP range query algorithm from Theorem B.1 on the input vector $(u_i(1), \dots, u_i(m))$ to get an output $(e^i_{\ell,r})_{1\leq\ell\leq r\leq m}$.
- For each allocation $A \in \mathcal{P}^{\text{conn}}(m, n)$, check whether $e^i_{\ell_i,r_i} \geq e^i_{\ell_{i'},r_{i'}} - 0.5c \cdot \gamma$ for all $i, i' \in N$, where $\{\ell_i, \dots, r_i\} = A_i$ and $\{\ell_{i'}, \dots, r_{i'}\} = A_{i'}$. If this holds, then output A.
- If the condition is not satisfied for any $A \in \mathcal{P}^{\text{conn}}(m, n)$, then output Null.

The running time of the algorithm follows from the fact that $|\mathcal{P}^{\text{conn}}(m, n)| \leq (mn)^n$. The algorithm is ε-DP due to the parallel composition theorem (Theorem 2.7) and the post-processing property of DP (Observation 2.6). For the utility, Theorem B.1 and the union bound ensures that with probability $1 - \beta$, we have $|e^i_{\ell,r} - u_i(\{\ell, \dots, r\})| \leq \zeta \cdot \log^{2.5}(mn/\beta)/\varepsilon$ for all $i \in N$ and $\ell, r \in M$. When this is the case, the EF2 allocation guaranteed by Theorem 2.1 passes the check. Furthermore, any allocation A that passes the check must be EFc. This is because, for all $i, i' \in N$, we have $u_i(A_{i'}) - u_i(A_i) \leq 0.5c \cdot \gamma + 2 \cdot \zeta \cdot \log^{2.5}(mn/\beta)/\varepsilon \leq c \cdot \gamma$; therefore, since i's

[17] Note that the guarantee stated in Theorem B.1 follows from Theorem 7.1 of Chan et al. [11] by plugging in $d = 1$ and applying the union bound across all $\leq m^2$ queries. (The "usefulness" definition of Chan et al. [11] only applies to each query, whereas our error notion is across all queries.)

utility for each item is either 0 or at least γ, removing the set S consisting of the c most valuable items for i from $A_{i'}$ results in $u_i(A_{i'} \setminus S) \le \max\{0, u_i(A_{i'}) - c \cdot \gamma\} \le u_i(A_i)$. This concludes our proof. □

Data availability

No data was used for the research described in the article.

References

[1] John M. Abowd, The US Census Bureau adopts differential privacy, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD), 2018, p. 2867.

[2] Georgios Amanatidis, Haris Aziz, Georgios Birmpas, Aris Filos-Ratsikas, Bo Li, Hervé Moulin, Alexandros A. Voudouris, Xiaowei Wu, Fair division of indivisible goods: recent progress and open questions, Artif. Intell. 322 (2023) 103965.

[3] Apple Differential Privacy Team, Learning with privacy at scale, http://machinelearning.apple.com/research/learning-with-privacy-at-scale, 2017. (Accessed 27 March 2023).

[4] Xiaohui Bei, Ayumi Igarashi, Xinhang Lu, Warut Suksompong, The price of connectivity in fair division, SIAM J. Discrete Math. 36 (2) (2022) 1156–1186.

[5] Xiaohui Bei, Alexander Lam, Xinhang Lu, Warut Suksompong, Welfare loss in connected resource allocation, in: Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI), 2024, pp. 2660–2668.

[6] Vittorio Bilò, Ioannis Caragiannis, Michele Flammini, Ayumi Igarashi, Gianpiero Monaco, Dominik Peters, Cosimo Vinci, William S. Zwicker, Almost envy-free allocations with connected bundles, Games Econ. Behav. 131 (2022) 197–221.

[7] Sylvain Bouveret, Yann Chevaleyre, Nicolas Maudet, Fair allocation of indivisible goods, in: Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, Ariel D. Procaccia (Eds.), Handbook of Computational Social Choice, Cambridge University Press, 2016, pp. 284–310, chapter 12.

[8] Sylvain Bouveret, Katarína Cechlárová, Edith Elkind, Ayumi Igarashi, Dominik Peters, Fair division of a graph, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI), 2017, pp. 135–141.

[9] Steven J. Brams, Alan D. Taylor, Fair Division: From Cake-Cutting to Dispute Resolution, Cambridge University Press, 1996.

[10] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D. Procaccia, Nisarg Shah, Junxing Wang, The unreasonable fairness of maximum Nash welfare, ACM Trans. Econ. Comput. 7 (3) (2019) 12:1–12:32.

[11] T.-H. Hubert Chan, Elaine Shi, Dawn Song, Private and continual release of statistics, ACM Trans. Inf. Syst. Secur. 14 (3) (2011) 26:1–26:24.

[12] Bhaskar Ray Chaudhury, Jugal Garg, Kurt Mehlhorn, EFX exists for three agents, J. ACM 71 (1) (2024) 4:1–4:27.

[13] Bolin Ding, Janardhan Kulkarni, Sergey Yekhanin, Collecting telemetry data privately, in: Proceedings of the 30th Annual Conference on Neural Information Processing Systems (NIPS), 2017, pp. 3571–3580.

[14] Lester E. Dubins, Edwin H. Spanier, How to cut a cake fairly, Am. Math. Mon. 68 (1) (1961) 1–17.

[15] Cynthia Dwork, Differential privacy: a survey of results, in: Proceedings of the 5th International Conference on Theory and Applications of Models of Computation (TAMC), 2008, pp. 1–19.

[16] Cynthia Dwork, Aaron Roth, The algorithmic foundations of differential privacy, Found. Trends Theor. Comput. Sci. 9 (3–4) (2014) 211–407.

[17] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, Moni Naor, Our data, ourselves: privacy via distributed noise generation, in: Proceedings of the 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT), 2006, pp. 486–503.

[18] Cynthia Dwork, Frank McSherry, Kobbi Nissim, Adam D. Smith, Calibrating noise to sensitivity in private data analysis, in: Proceedings of the 3rd Theory of Cryptography Conference (TCC), 2006, pp. 265–284.

[19] Cynthia Dwork, Moni Naor, Omer Reingold, Guy N. Rothblum, Salil P. Vadhan, On the complexity of differentially private data release: efficient algorithms and hardness results, in: Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC), 2009, pp. 381–390.

[20] Cynthia Dwork, Moni Naor, Toniann Pitassi, Guy N. Rothblum, Differential privacy under continual observation, in: Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC), 2010, pp. 715–724.

[21] Úlfar Erlingsson, Vasyl Pihur, Aleksandra Korolova, Rappor: randomized aggregatable privacy-preserving ordinal response, in: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS), 2014, pp. 1054–1067.

[22] Even Shimon, Azaria Paz, A note on cake cutting, Discrete Appl. Math. 7 (3) (1984) 285–296.

[23] Badih Ghazi, Ravi Kumar, Pasin Manurangsi, Thomas Steinke, Algorithms with more granular differential privacy guarantees, in: Proceedings of the 14th Innovations in Theoretical Computer Science Conference (ITCS), 2023, pp. 54:1–54:24.

[24] Andy Greenberg, Apple's 'differential privacy' is about collecting your data—but not *your* data, http://www.wired.com/2016/06/apples-differential-privacy-collecting-data, 2016. (Accessed 27 March 2023).

[25] Moritz Hardt, Kunal Talwar, On the geometry of differential privacy, in: Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC), 2010, pp. 705–714.

[26] Justin Hsu, Zhiyi Huang, Aaron Roth, Tim Roughgarden, Zhiwei Steven Wu, Private matchings and allocations, SIAM J. Comput. 45 (6) (2016) 1953–1984.

[27] Ayumi Igarashi, How to cut a discrete cake fairly, in: Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI), 2023, pp. 5681–5688.

[28] Sampath Kannan, Jamie Morgenstern, Ryan Rogers, Aaron Roth, Private Pareto optimal exchange, ACM Trans. Econ. Comput. 6 (3–4) (2018) 12:1–12:25.

[29] David T. Lee, Efficient, private, and ϵ-strategyproof elicitation of tournament voting rules, in: Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI), 2015, pp. 2026–2032.

[30] Zhechen Li, Ao Liu, Lirong Xia, Yongzhi Cao, Hanpin Wang, Differentially private Condorcet voting, in: Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI), 2023, pp. 5755–5763.

[31] Richard J. Lipton, Evangelos Markakis, Elchanan Mossel, Amin Saberi, On approximately fair allocations of indivisible goods, in: Proceedings of the 5th ACM Conference on Economics and Computation (EC), 2004, pp. 125–131.

[32] Pasin Manurangsi, Warut Suksompong, Almost envy-freeness for groups: improved bounds via discrepancy theory, Theor. Comput. Sci. 930 (2022) 179–195.

[33] Pasin Manurangsi, Warut Suksompong, Differentially private fair division, in: Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI), 2023, pp. 5814–5822.

[34] Evangelos Markakis, Approximation algorithms and hardness results for fair division with indivisible goods, in: Ulle Endriss (Ed.), Trends in Computational Social Choice, AI Access, 2017, pp. 231–247, chapter 12.

[35] Frank McSherry, Privacy integrated queries: an extensible platform for privacy-preserving data analysis, Commun. ACM 53 (9) (2010) 89–97.

[36] Frank McSherry, Kunal Talwar, Mechanism design via differential privacy, in: Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 2007, pp. 94–103.

[37] Hervé Moulin, Fair Division and Collective Welfare, MIT Press, 2003.

[38] Hervé Moulin, Fair division in the Internet age, Annu. Rev. Econ. 11 (2019) 407–441.

[39] Trung Thanh Nguyen, Jörg Rothe, Complexity results and exact algorithms for fair division of indivisible items: a survey, in: Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI), 2023, pp. 6732–6740.

[40] Herbert Robbins, A remark on Stirling's formula, Am. Math. Mon. 62 (1) (1955) 26–29.
[41] Jack Robertson, William Webb, Cake-Cutting Algorithms: Be Fair If You Can, Peters/CRC Press, 1998.
[42] Shang Shang, Tiance Wang, Paul Cuff, Sanjeev Kulkarni, The application of differential privacy for rank aggregation: privacy and accuracy, in: Proceedings of the 17th International Conference on Information Fusion (FUSION), 2014.
[43] Stephen Shankland, How Google tricks itself to protect chrome user privacy, http://www.cnet.com/tech/services-and-software/how-google-tricks-itself-to-protect-chrome-user-privacy, 2014. (Accessed 27 March 2023).
[44] Thomas Steinke, Jonathan R. Ullman, Between pure and approximate differential privacy, J. Priv. Confid. 7 (2) (2016) 3–22.
[45] Warut Suksompong, Fairly allocating contiguous blocks of indivisible items, Discrete Appl. Math. 260 (2019) 227–236.
[46] Warut Suksompong, Constraints in fair division, ACM SIGecom Exch. 19 (2) (2021) 46–61.
[47] Warut Suksompong, Weighted fair division of indivisible items: a review, Inf. Process. Lett. 187 (2025) 106519.
[48] Salil Vadhan, The complexity of differential privacy, in: Yehuda Lindell (Ed.), Tutorials on the Foundations of Cryptography, Springer International Publishing, 2017, pp. 347–450, chapter 7.
[49] Toby Walsh, Fair division: the computer scientist's perspective, in: Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI), 2020, pp. 4966–4972.