Leximin Allocations in the Real World

DAVID KUROKAWA and ARIEL D. PROCACCIA, Carnegie Mellon University, USA
NISARG SHAH, University of Toronto, Canada

As part of a collaboration with a major California school district, we study the problem of *fairly* allocating unused classrooms in public schools to charter schools. Our approach revolves around the randomized *leximin mechanism*. We extend previous work to show that the leximin mechanism is proportional, envy-free, Pareto optimal, and group strategyproof, not only in our classroom allocation setting, but in a general framework that subsumes a number of settings previously studied in the literature. We also prove that the leximin mechanism provides a (worst-case) 4-approximation to the maximum number of classrooms that can possibly be allocated. Our experiments, which are based on real data, show that a non-trivial implementation of the leximin mechanism scales gracefully in terms of running time (even though the problem is intractable in theory), and performs extremely well with respect to a number of efficiency objectives. We establish the practicability of our approach, and discuss issues related to its deployment.

CCS Concepts: • **Theory of computation → Design and analysis of algorithms**; • **Computing methodologies → Artificial intelligence**; • **Applied computing → Economics**;

Additional Key Words and Phrases: Fair division, random assignment, leximin

ACM Reference format:
David Kurokawa, Ariel D. Procaccia, and Nisarg Shah. 2018. Leximin Allocations in the Real World. *ACM Trans. Econ. Comput.* 6, 3–4, Article 11 (October 2018), 24 pages.
https://doi.org/10.1145/3274641

1 INTRODUCTION

Over the course of the last seven decades, the study of fair division has given rise to a slew of elegant solutions to a variety of problems [9, 24, 32], which span the practicability spectrum from abstract (e.g., cake cutting [30]) to everyday (e.g., rent division [1]). Building on its rich history, the field of fair division—and *computational* fair division, in particular—is poised to make a significant impact on society through applications that are beginning to emerge. For example, Budish's fair division approach [10]—which leads to challenging computational questions [26, 27]—is now regularly used by the Wharton School of the University of Pennsylvania to allocate seats in MBA courses. And the not-for-profit website *Spliddit* (`www.spliddit.org`—which offers provably fair

A preliminary version of this article appeared in Proceedings of the 16th ACM Conference on Economics and Computation, pp. 345–362, 2015.

This work was partially supported by NSF grants CCF-1215883 and IIS-1350598, by a Sloan Research Fellowship, and by the NSERC under the Discovery Grants program.

Authors' addresses: D. Kurokawa, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213; email: dkurokaw@cs.cmu.edu; A. D. Procaccia, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213; email: arielpro@cs.cmu.edu; N. Shah, University of Toronto, 10 King's College Rd., Toronto, ON M5G2R3, Canada; email: nisarg@cs.toronto.edu.

2167-8375/2018/10-ART11 $15.00
https://doi.org/10.1145/3274641

solutions for the division of goods, rent, and credit [19]—has already been used by more than 100,000 people.

One of the beautiful consequences of these applications is that people have become aware of fair division theory, and are reaching out with problems that are possibly specialized, yet just as significant in terms of societal impact. This was foreseen by the esteemed economist Hervé Moulin, who wrote to one of us (Procaccia) by email on June 3, 2013 (in the context of Spliddit's early development):

> *"I believe that, with few exceptions (school choice?) academics like us are not going to invent from their armchair the best applications of our models, concepts and solutions, although we have a good sense of the type of problems where they can help. Thus the reward of helping people who have a real fair division problem by explaining our solutions, is that they in return pose interesting and difficult new questions, food for our thoughts. So if the website lets users ask questions of their own, it could be a goldmine of ideas, as well as a costly proposition if there are too many questions!"*

This article presents a solution to one of these "interesting and difficult new questions," posed by a representative of one of the largest school districts in California. Since the details are confidential, we will refer to the school district as the Pentos Unified School District (PUSD), and to the representative as Mr. Mopatis. Mr. Mopatis contacted us in May 2014 after learning about Spliddit (and fair division, more generally) from an article in the New York Times.[1] He is tasked with the allocation of unused space (most importantly, classrooms) in PUSD's public schools to the district's charter schools, according to California's Proposition 39, which states that "public school facilities should be shared fairly among all public school pupils, including those in charter schools."[2] While the law does not elaborate on what "fairly" means, Mr. Mopatis was motivated by the belief that a provably fair solution would certainly fit the bill. He asked us to design an automated allocation method that would be evaluated by PUSD, and potentially replace the existing manual system.

To be a bit more specific, the setting consists of charter schools and facilities (public schools). Each facility has a given number of unused classrooms—its *capacity*, and each charter school has a number of required classrooms—its *demand*. In principle, the classrooms required by a charter school could be split across multiple facilities, but such offers have always been declined in the past, so we assume that an agent's demand must be satisfied in a single facility (if it is satisfied at all). Other details are less important and can be abstracted away. For example, classroom size turns out to be a non-issue, and the division of time in shared space (such as the school gym or cafeteria) can be handled ad hoc.

Of course, to talk of fairness we must also take into account the preferences of charter schools, but preference representation is a modeling choice, intimately related to the design and guarantees of the allocation mechanism. Moreover, fairness is not our only concern: to be used in practice, the mechanism must be relatively intuitive (so it can be explained to decision makers) and computationally feasible. The challenge we address is therefore to

> *... design and implement a classroom allocation mechanism that is provably fair as well as practicable.*

1.1 Our Approach and Results

We model the preferences of charter schools as being *dichotomous*: charter schools think of each facility as either acceptable or unacceptable. This choice is motivated by current practice: Under

[1] http://goo.gl/Xp3omV.
[2] http://goo.gl/bGH6dT.

the 2015/2016 request form issued by PUSD, charter schools are essentially asked to indicate acceptable facilities (specifically, they are asked to "provide a description of the district school site and/or general geographic area in which the charter school wishes to locate" using free-form text). In other words, formally eliciting dichotomous preferences—by having charter schools select acceptable facilities from the list of all facilities—is similar to the *status quo*, a fact that increases the practicability of the approach.

A natural starting point, therefore, is the seminal paper of Bogomolnaia and Moulin [7], who study the special case of our setting with unit demands and capacities, under dichotomous preferences. They propose the *leximin mechanism*, which returns a random allocation with the following intuitive property: it maximizes the lowest probability of any charter school having its demand satisfied in an acceptable facility; subject to this constraint, it maximizes the second lowest probability; and so on.

In Section 3, we show that the leximin mechanism remains compelling in the classroom allocation setting. Specifically, we prove that it satisfies the following properties: (i) proportionality—each charter school receives its proportional share of available classrooms; (ii) envy-freeness—each charter school prefers its own allocation to the allocation of any other school; (iii) Pareto optimality (a.k.a. ex-ante efficiency)—no other randomized allocation is at least as good for all charter schools, and strictly better for at least one; and (iv) group strategyproofness—even coalitions of charter schools cannot benefit by misreporting their preferences. The beauty of these properties, as well as the leximin mechanism itself, is that they are intuitive and can easily be explained to a layperson. This feature, once again, significantly contributes to the practicability of the approach. As an interesting aside, we show that the leximin mechanism still satisfies the foregoing properties in a much more general setting, thereby generalizing results from a variety of other papers in fair division and mechanism design.

In Section 4, we study the leximin mechanism from a combinatorial optimization viewpoint. The section's main result is that the expected number of classrooms allocated by the leximin mechanism is always at least 1/4 of the maximum number of classrooms that can possibly be allocated. We do not view this result as enhancing the practicability of our approach, but rather as significantly contributing to its intellectual merit. We further conjecture that an improved bound of 1/2 is feasible.

In Section 5, we observe that the problem of computing a leximin allocation is $\mathcal{NP}$-hard in our setting, and describe our implementation of the leximin mechanism—a task which has proved quite challenging. A naïve approach to the computation of leximin allocations solves a sequence of linear programs, each with an exponential number of variables. On a high level, the crux of our implementation is that we work with the duals of these linear programs—each with an exponential number of constraints—and formulate a separation oracle as an integer linear program.

Finally, in Section 6, we present our experiments. Using an instance generator that is grounded in historical data from PUSD, we show that our algorithm for computing leximin allocations scales quite gracefully. In particular, even when there are 300 charter schools (which is more than any school district in the U.S. has), the algorithm terminates in a few minutes on average. Remarkably, we also observe that, in our experiments, the leximin mechanism satisfies (on average) at least 98% of the maximum number of charter schools that can possibly be satisfied, and allocates (on average) at least 98% of the maximum number of classrooms that can possibly be allocated.

1.2 Related Work

The problem of fairly dividing a set of indivisible goods has been studied extensively. As an early, seminal example, Hylland and Zeckhauser [20] propose a compelling pseudo-market mechanism to compute a lottery over deterministic assignments, given cardinal preferences. Their mechanism satisfies proportionality, envy-freeness, and ex-ante efficiency, but fails to provide strategyproofness. A more serious objection to their mechanism is that they elicit cardinal utilities from agents—a difficult task in practice. A market approach also drives the work of [10] on approximate competitive equilibrium from equal incomes. His approximation guarantees are practical as long as the supply of each good is relatively large, which is not the case in the classroom allocation setting (where the number of available classrooms in a facility is typically small).

Bogomolnaia and Moulin [6] study random assignment under ordinal preferences. They introduce the *probabilistic serial* (PS) mechanism, which satisfies ex-ante efficiency as well as ordinal fairness. Informally, the probabilistic serial mechanism allows agents to "eat" (at identical speeds) their shares of different goods one by one in the order in which they rank the goods. However, similarly to the pseudo-market mechanism of Hylland and Zeckhauser [20], the probabilistic serial mechanism pertains to the basic setting of assigning n indivisible goods to n agents.

Budish et al. [11] propose a general framework, which, by generalizing the classic Birkhoff von-Neumann theorem [4, 36], extends both mechanisms to handle real-world combinatorial domains, e.g., with group quotas, endogenous capacities, multi-unit non-additive demands, scheduling constraints, and so on. Their extension of the probabilistic serial mechanism would be a potential starting point in our setting, if we wished to elicit ordinal preferences from the agents. However, note that in our setting a charter school demanding d classrooms must either receive all d classrooms at a single facility or no classrooms at all—this restriction is incompatible with the framework of [11]. There are other extensions of the probabilistic serial mechanism with multi-unit demands [3, 12, 22, 31], but all of them leverage the standard Birkhoff von-Neumann theorem to allocate *at most* d goods to an agent, and cannot ensure that the agent receives exactly d goods (or no goods at all). We consider it an interesting open problem to extend the probabilistic serial mechanism to the classroom allocation setting with ordinal preferences.

As noted above, Bogomolnaia and Moulin [7] show that if we move to a setting with dichotomous preferences, much stronger guarantees can be provided. In particular, they show that for the classic setting with n agents and n goods the leximin mechanism satisfies proportionality, envy-freeness, Pareto optimality, and strategyproofness. We generalize (some of) their results by proving that the leximin mechanism satisfies these four properties in our setting as well. Other properties established by [7], such as the Lorenz dominance of the leximin probability vector, do not hold in our setting (as we demonstrate below).

[8] study a more general dichotomous preferences setting where every agent essentially accepts a subset of feasible deterministic allocations.[3] They propose the utilitarian mechanism, which uniformly randomizes over all deterministic allocations maximizing social welfare, and show that it satisfies envy-freeness, Pareto optimality, and strategyproofness, but violates proportionality and suffers from "tyranny of the majority."[4] This makes the mechanism highly undesirable in our setting; see the discussion in Section 4.

2 THE MODEL

We begin by formalizing the classroom allocation setting that motivates our work. Let $N = \{1, \ldots, n\}$ denote the set of charter schools (hereinafter, *agents*), and let $M = \{1, \ldots, m\}$ denote the set of public schools (hereinafter, *facilities*). We want to design a mechanism for assigning the agents to the facilities. Each facility f has a *capacity* c_f, which is the number of *units* available at

[3] As we discuss in Section 3.2, this general dichotomous preference setting captures our classroom allocation setting, but our proofs extend to an extremely general setting that actually captures the general dichotomous preference setting as a special case.

[4] Apparently, this result was later independently discovered by Freitas [15].

the facility (in our motivating example, each unit is a classroom). The preferences of agent i are given by a pair (d_i, F_i), where $d_i \in \mathbb{N}$ denotes the number of units demanded by agent i—or, simply, the *demand* of agent i—and $F_i \subseteq M$ denotes the set of facilities acceptable to agent i. Crucially, we assume that agent i's preferences are dichotomous in nature: the agent has utility 1 if it receives d_i units from *any single facility* $f \in F_i$ (in this case, we say agent i is assigned to facility f), and 0 otherwise. Without loss of generality, we assume that every agent i has an acceptable facility $f \in F_i$ that has sufficient capacity to meet its demand (i.e., $c_f \geq d_i$).[5]

A deterministic allocation is a mapping $A : N \to M \cup \{0\}$, where $A_i = A(i)$ denotes the facility to which agent i is assigned (and $A_i = 0$ means agent i is not assigned to any facility). A is *feasible* if it respects the capacity constraint at each facility:

$$\forall f \in M, \sum_{i \in N : A_i = f} d_i \leq c_f.$$

Let $\mathcal{A}$ denote the space of all feasible deterministic allocations. Formally, the utility to agent i under a feasible deterministic allocation $A \in \mathcal{A}$ is given by

$$u_i(A_i) = \begin{cases} 1 & \text{if } A_i \in F_i \\ 0 & \text{otherwise.} \end{cases}$$

A feasible randomized allocation is simply a distribution over feasible deterministic allocations, and the utility to an agent is its expected utility under the randomized allocation. Let $\Delta(\mathcal{A})$ be the space of all feasible randomized allocations. Crucially, note that $\Delta(\mathcal{A})$ is a convex set, i.e., given randomized allocations $A, A' \in \Delta(\mathcal{A})$ and $0 \leq \lambda \leq 1$, we can construct another randomized allocation $A'' = \lambda \cdot A + (1 - \lambda) \cdot A' \in \Delta(\mathcal{A})$ that executes A with probability λ and A' with probability $1 - \lambda$. Hereinafter, an allocation is possibly randomized, unless explicitly specified otherwise.

As mentioned in Section 1, our setting deals with fair allocation of indivisible goods, and generalizes the classic setting of random assignment under dichotomous preferences studied by Bogomolnaia and Moulin [7]. In particular, their setting can be recovered by setting all the demands and capacities to 1 (i.e., $d_i = 1$ and $c_f = 1$ for all $i \in N$, $f \in M$), with an equal number of agents and facilities ($m = n$).

Desiderata. The fair division literature offers a slew of desirable properties. We are especially interested in four classic desiderata that have proved to be widely applicable (with applications ranging from cake cutting [30] to the division of computational resources in clusters [18, 28]), often satisfiable, and yet effective in leading to compelling mechanisms. We use these desiderata to guide the search for a good mechanism in our setting. Let A denote an allocation returned by a mechanism under consideration.

(1) *Proportionality.* This is a fairness requirement that states that every agent should receive at least its proportional share of the available goods. Since the maximum utility any agent can achieve is 1, a mechanism is called proportional if the utility to each agent is at least $1/n$, i.e., if $u_i(A_i) \geq 1/n$ for all $i \in N$.
(2) *Envy-Freeness.* This is another fairness requirement which states that every agent should prefer its own allocation over the allocation of any other agent. In other words, no agent should envy any other agent. Formally, a mechanism is called envy-free if $u_i(A_i) \geq u_i(A_j)$ for all $i, j \in N$.
(3) *Pareto Optimality.* This is a qualitative notion of efficiency which requires that it be impossible to make an agent better off without making some other agent worse off. Formally,

[5] Agents violating this requirement can never achieve positive utility, and can effectively be disregarded.

an allocation A is Pareto dominated by an allocation A' (or A' is a Pareto improvement over A) if $u_i(A'_i) \geq u_i(A_i)$ for every agent $i \in N$ and $u_i(A'_i) > u_i(A_i)$ for some agent $i \in N$. A mechanism is called Pareto optimal if the allocation it returns is not Pareto dominated by any alternative allocation. In our context, Pareto optimality denotes ex-ante efficiency, which is a *strictly* stronger notion than ex-post efficiency, as the latter notion only requires an allocation to be a randomization over deterministic Pareto optimal allocations.

(4) *Strategyproofness.* This property is a strong game-theoretic requirement. In our setting, the preferences of agent i (both d_i and F_i) are its private information. We would like to motivate each agent to report its preferences truthfully regardless of the preferences reported by the other agents. A mechanism is called strategyproof if truth-telling is a dominant strategy for every agent. Formally, let A denote the allocation returned when the preferences reported by the agents are $(\mathbf{d}, \mathbf{F})$, and let A' denote the allocation returned when an agent $i \in N$ changes its preferences to (d'_i, F'_i) while the preferences of the other agents remain unchanged. Then, we require that $u_i(A_i) \geq u_i(A'_i)$, where u_i is the utility function induced by the original preferences (d_i, F_i). A stronger notion called *group strategyproofness* requires that if a subset of agents simultaneously report false preferences, at least one of the agents in the subset must not be strictly better off.

Let us first consider an example illustrating the desiderata of our interest.

Example 1. First, let us consider a simple randomized mechanism that allocates all available units at all facilities to each agent with probability $1/n$. Clearly, the mechanism satisfies proportionality because it gives each agent utility $1/n$. The mechanism is also envy-free because each agent has an identical allocation, and thus no reason to envy any other agent. Since the mechanism operates independently of the reported preferences of the agents, the mechanism is obviously (group) strategyproof. However, the mechanism is not Pareto optimal. The reason is that the mechanism allocates all available units to an agent (with probability $1/n$) even if the agent does not require all the units. In this case, it may be possible to simultaneously satisfy another agent, thus obtaining a Pareto improvement.

Next, consider a different mechanism that always returns a deterministic allocation maximizing the number of units allocated. While this mechanism is very intuitive, we can show that it violates all the desiderata except Pareto optimality. Suppose there is a single facility with four available units, and two agents—namely, agents 1 and 2—that demand three and two units, respectively. Maximizing the number of units allocated would require allocating three units to agent 1 and no units to agent 2. This already violates both proportionality and envy-freeness with respect to agent 2. Further, agent 2 would have a strict incentive to report a false demand of four units, which would lead to agent 2 receiving all four units from the facility. Thus, strategyproofness is also violated.

3 THE LEXIMIN MECHANISM

Let us consider the leximin mechanism proposed by Bogomolnaia and Moulin [7] (for the special case of random assignment under dichotomous preferences) in our more general setting. Informally, the leximin mechanism first maximizes the minimum utility that any agent achieves. Then, subject to this constraint, it maximizes the second lowest utility, and so on. Formally, let $(u^1, u^2, \ldots, u^n)$ denote the vector of utilities sorted in non-descending order. The leximin mechanism returns the allocation that maximizes this vector in the lexicographic order; we say that this allocation is leximin-optimal. The mechanism is presented as Algorithm 1. Note that Algorithm 1 is highly intractable in practice; we describe a practical implementation of the leximin mechanism in Section 5.

ALGORITHM 1: The Leximin Mechanism

Data: Demands $\{(d_i, F_i)\}_{i \in N}$, Capacities $\{c_f\}_{f \in M}$
Result: The Leximin-Optimal Allocation A
For $k \in \{1, \ldots, n\}$, let u^k denote the k^{th} lowest utility under an allocation;
for $k = 1$ *to* n **do**
  $\bar{u}^k \leftarrow$ Max u^k subject to $u^j = \bar{u}^j$ for all $j < k$;
end
return an allocation where $u^k = \bar{u}^k$ for all $k \in \{1, \ldots, n\}$;

In a sense, the leximin mechanism is an extension of the egalitarian equivalence principle put forward by Pazner and Schmeidler [29], in which one attempts to equalize all agent utilities (and maximize this utility value). This is what the leximin mechanism attempts in its first step of maximizing the minimum utility. However, sometimes the solution obtained is not Pareto optimal. The subsequent steps amend this solution to make it Pareto optimal, and eliminate any waste of resources. Without loss of generality, assume that the leximin mechanism chooses a non-wasteful allocation, i.e., under every deterministic assignment in its support agent i either receives d_i units from a facility in F_i or does not receive any units. Let us illustrate how the leximin mechanism works through an example.

Example 2. Suppose there are two facilities a and b with capacities $c_a = 1$ and $c_b = 2$, respectively, and four agents with demands $d_1 = 1$, $d_2 = 1$, $d_3 = 1$, and $d_4 = 2$. Suppose agent 1 only accepts facility a ($F_1 = \{a\}$), agent 2 accepts both facilities ($F_2 = \{a, b\}$), and agents 3 and 4 only accept facility b ($F_3 = F_4 = \{b\}$). This is shown in Figure 1(a), where the agents are shown on the left with their demands, the facilities are shown on the right with their capacities, and each agent is connected to each of its acceptable facilities through a dashed line.

It is clear that the minimum utility cannot be greater than 1/2 because agents 3 and 4 must be assigned to facility b separately. The allocation shown in Figure 1(b) gives utility 1/2 to all agents. However, this is not sufficient for the allocation to be the leximin allocation. For instance, the allocation shown in Figure 1(c) increases the utility to agent 2 while preserving the utilities to the other agents, and is therefore better in a lexicographic comparison of the sorted utility vector. While this new allocation is Pareto optimal, it is still not the leximin allocation. The leximin allocation is shown in Figure 1(d)—it gives utility 1/2 to agents 3 and 4, and utility 3/4 to agents 1 and 2. This achieves the same lowest and second lowest utilities as the previous two allocations, but a greater third lowest utility than both previous allocations.

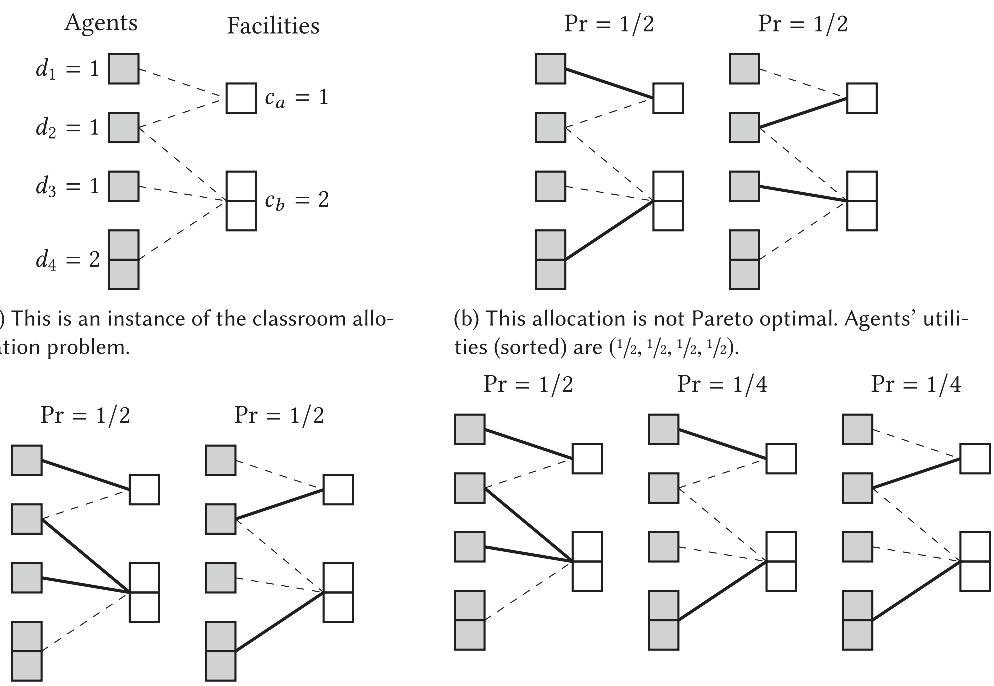
3.1 Properties of the Leximin Mechanism

Bogomolnaia and Moulin [7] show that the leximin mechanism satisfies all four desiderata proposed above in their classic setting with one-to-one matchings, and unit demands and capacities. We now show that these properties continue to hold in our setting with many-to-one matchings, and arbitrary demands and capacities. In fact, in Section 3.2 we argue that they hold in an even more general setting.

Theorem 3. *The leximin mechanism satisfies proportionality, envy-freeness, Pareto optimality, and group strategyproofness.*

Proof. We first formally establish an intuitive property of leximin allocations. □

Lemma 4. *Let A denote the allocation returned by the leximin mechanism. Then for utility function u induced by any dichotomous preferences, we have $u_i(A_i) \geq u(A_i)$.*



(a) This is an instance of the classroom allocation problem.

(b) This allocation is not Pareto optimal. Agents' utilities (sorted) are ($1/2, 1/2, 1/2, 1/2$).

(c) This is Pareto optimal, but not the leximin allocation. Agents' utilities (sorted) are ($1/2, 1/2, 1/2, 1$).

(d) This allocation is the leximin allocation (and is therefore Pareto optimal). Agents' utilities (sorted) are ($1/2, 1/2, 3/4, 3/4$).

Fig. 1. Illustration of Example 2.

Proof. First, let A be deterministic. If $A_i \neq 0$, then due to the non-wastefulness of the leximin allocation, we must have $u_i(A_i) = 1 \geq u(A_i)$ for any utility function u. On the other hand, $A_i = 0$ implies $u_i(A_i) = u(A_i) = 0$ for all utility functions u. Hence, the lemma holds for all deterministic allocations. For randomized allocations, taking expectation on both sides yields that the lemma still holds. (Proof of Lemma 4) □

Proportionality. Consider the mechanism that allocates all available units to each agent with probability $1/n$, which gives each agent utility $1/n$.[6] Since the leximin mechanism maximizes the minimum utility that any agent receives, it must also give each agent at least $1/n$ utility. Hence, the leximin mechanism is proportional.

Envy-Freeness. Suppose for contradiction that under an allocation A returned by the leximin mechanism, agent i envies agent j. That is, $u_i(A_j) > u_i(A_i)$. Now, Lemma 4 implies $u_j(A_j) \geq u_i(A_j) > u_i(A_i) \geq 0$. Let $0 < \epsilon < (u_j(A_j) - u_i(A_i))/u_j(A_j)$.

Construct another allocation A' such that $A'_k = A_k$ for all $k \in N \setminus \{i, j\}$, $A'_i = A_j$, and $A'_j = 0$. Since agent i envied agent j, we have $d_i \leq d_j$, implying that A' is feasible. Note that agent i now has higher utility because $u_i(A'_i) = u_i(A_j) > u_i(A_i)$.

Construct an allocation A'' that realizes A with probability $1 - \epsilon$ and A' with probability ϵ. Due to our construction of A'', we have that for every agent $k \in N \setminus \{i, j\}$, $u_k(A'') = u_k(A') = u_k(A)$.

[6]This is because we assumed that the demand of every agent can be satisfied given all available units.

Further, for agent i we have $u_i(A''_i) > u_i(A_i)$. Also, for agent j we have

$$u_j(A''_j) = (1-\epsilon)u_j(A_j) > u_i(A_i).$$

Hence, switching from A to A'' preserves the utility achieved by every agent except agents i and j, and both agents i and j receive utility strictly greater than $u_i(A_i) = \min(u_i(A_i), u_j(A_j))$. That is, allocation A'' is strictly better than allocation A in the leximin ordering, which contradicts the leximin-optimality of A.

Pareto Optimality. This follows trivially from the definition of leximin-optimality. Note that increasing the utility of an agent i without decreasing the utility of any other agent would improve the allocation in the leximin ordering. Since the allocation returned by the leximin mechanism is already leximin-optimal, it does not admit any Pareto improvements. Hence, the leximin mechanism is Pareto optimal.

Group Strategyproofness. This is the most non-trivial property to establish among the four desired properties. Under the true reports $(d_k, F_k)_{k\in N}$, let A denote the allocation returned by the leximin mechanism. Suppose a subset of agents $S \subseteq N$, whom we call manipulators, report false preferences $(d'_i, F'_i)_{i\in S}$; let $(u'_i)_{i\in S}$ denote the utility functions induced by the false preferences of the manipulators. Let A' denote the allocation returned by the leximin mechanism when agents in S misreport. Suppose for contradiction that every agent in S is strictly better off (under their true utility functions) by misreporting, i.e., $u_i(A'_i) > u_i(A_i)$ for every $i \in S$. Now, Lemma 4 implies that $u'_i(A'_i) \geq u_i(A'_i)$; thus, we have $u'_i(A'_i) > u_i(A_i)$ for every $i \in S$.

Before we derive a contradiction, we first observe that the leximin-optimality of an allocation implies Pareto optimality of any prefix of its sorted utility vector. Let $\text{pref}_A(i) = \{j \in N \mid u_j(A_j) \leq u_i(A_i)\}$ denote the prefix of agent i in allocation A.

Lemma 5 (Prefix Optimality). *For an allocation X returned by the leximin mechanism and an agent $i \in N$, there does not exist an allocation X' such that some agent in $\text{pref}_X(i)$ is strictly better off under X' and no agent in $\text{pref}_X(i)$ is worse off.*

Proof. Assume without loss of generality that $u_i(X_i) < \max_{j\in N} u_j(X_j)$, otherwise the statement coincides with Pareto optimality. Suppose for contradiction that an allocation X' as in the statement of the lemma exists. Choose ϵ such that

$$0 < \epsilon < \frac{1 - u_i(X_i)}{\min\{u_j(X_j) \mid u_j(X_j) > u_i(X_i)\}}.$$

Consider the allocation $X'' = (1-\epsilon)\cdot X + \epsilon \cdot X'$. Due to our choice of ϵ, we can see that for every agent $j \notin \text{pref}_X(i)$, we have $u_j(X''_j) \geq (1-\epsilon)u_j(X_j) > u_i(X_i)$. Further, we have $u_j(X''_j) \geq u_j(X_j)$ for every agent $j \in \text{pref}_X(i)$ and $u_j(X''_j) > u_j(X_j)$ for some $j \in \text{pref}_X(i)$.

We now show that X'' is strictly better than X in the leximin ordering. Choose agent $j^* \in arg\,min_{j\in \text{pref}_X(i):u_j(X''_j)>u_j(X_j)} u_j(X_j)$. Break ties by choosing an agent with the smallest value of $u_j(X''_j)$, and if there are still ties, break them arbitrarily. Let $t = |\{k \in \text{pref}_X(i) | u_k(X_k) < u_{j^*}(X_{j^*})\}| + |\{k \in \text{pref}_X(i) | u_k(X''_k) = u_k(X_k) = u_j(X_j)\}|$. Then, one can check that allocations X and X'' match in the t lowest utilities, and allocation X'' has a strictly greater $(t+1)^{st}$ lowest utility. Thus, X'' is strictly better than X in the leximin ordering, which contradicts leximin-optimality of X. (Proof of Lemma 5) □

Fix a manipulator $i \in S$ that minimizes $u_i(A_i)$ among all $i \in S$ (break ties arbitrarily). Let us look at the set of all agents that are strictly better off under A' compared to A, and among these agents, choose an agent j that minimizes $u_j(A_j)$ (again, break ties arbitrarily). Now, agent i is also strictly better off under A'. Hence, by the definition of agent j, we have $u_j(A_j) \leq u_i(A_i)$. Since agent j is strictly better off under A', by prefix optimality of A (Lemma 5) we know there must exist an agent in $\text{pref}_A(j)$ that is strictly worse off under A'. Among all agents in $\text{pref}_A(j)$ that are worse off under A', choose an agent k that minimizes $u_k(A'_k)$ (again, break ties arbitrarily).

Now, we derive our contradiction by showing that prefix optimality of A' is violated. More precisely, we know that agent k is strictly better off under A compared to A'. We show that no agent in $\text{pref}_{A'}(k)$ is worse off under A compared to A'.

First, note that for any manipulator $l \in S$, we have $u_{l'}(A'_l) \geq u_l(A'_l) > u_l(A_l) \geq u_i(A_i) \geq u_j(A_j) \geq u_k(A_k) > u_k(A'_k)$, where the third, fourth, and fifth transitions hold due to our choice of agents i, j, and k, respectively. Thus, no manipulator belongs to $\text{pref}_{A'}(k)$. In other words, for every agent $l \in \text{pref}_{A'}(k)$ we can denote its utility function (which is common between A and A') by u_l. Take an agent $l \in \text{pref}_{A'}(k)$. If $u_l(A_l) < u_l(A'_l)$, then we have $u_l(A_l) < u_l(A'_l) \leq u_k(A'_k) < u_k(A_k) \leq u_j(A_j)$. Thus, agent l satisfies $u_l(A_l) < u_j(A_j)$, and is still better off under A' compared to A, which contradicts our choice of agent j. Therefore, $u_l(A_l) \geq u_l(A'_l)$ for every $l \in \text{pref}_{A'}(k)$, and $u_k(A_k) > u_k(A'_k)$, contradicting prefix optimality of A'. (Proof of Theorem 3) □

While group strategyproofness is a strong game-theoretic requirement, an even stronger requirement has been studied in the literature. Under this stronger requirement, a group of manipulators should not be able to report false preferences that would lead to all manipulators being weakly happier and at least one manipulator being strictly happier. Bogomolnaia and Moulin [7] show that in the classical random assignment setting under dichotomous preferences, the leximin mechanism is group strategyproof according to this stronger requirement.[7] Unfortunately, the following example shows that this does not hold in our more general setting.

Example 6. Suppose there are nine agents with demands

$$(d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9) = (2, 4, 4, 4, 2, 2, 2, 1, 1),$$

and three facilities with capacities $(c_1, c_2, c_3) = (4, 2, 1)$. Let the dichotomous preferences of the agents be as follows: $F_i = \{1\}$ for $i \in \{1, 2, 3, 4\}$, $F_5 = \{1, 2\}$, $F_6 = F_7 = \{2\}$, $F_8 = \{2, 3\}$, and $F_9 = \{3\}$.

In this case, it can be checked that under the leximin allocation, the utilities of the agents are as follows: $u_i = 1/4$ for $i \in \{1, 2, 3, 4\}$, $u_5 = u_6 = u_7 = 5/12$, and $u_8 = u_9 = 1/2$.

Suppose agent 1 manipulates, and increases its demand to $d'_1 = 3$ units. Then, it can be checked that under the new leximin allocation, the utility of agents 1 through 4 remains 1/4, the utility of agents 5 through 7 drops to 1/3, and the utility of agents 8 and 9 increases to 5/8. Thus, agent 1 and agent 9 form a successful group manipulation in which no agent is worse off, but agent 9 is strictly better off.

Similarly, Bogomolnaia and Moulin [7] also show that a leximin-optimal allocation always *Lorenz-dominates* any other allocation in their classic setting. Let us first define Lorenz dominance among allocations.

Lorenz Dominance. For $k \in \{1, \ldots, n\}$, let u^k and v^k denote the k^{th} lowest utility in allocations A and B, respectively. We say that allocation A (weakly) Lorenz-dominates allocation B if $\sum_{i=1}^k u^i \geq \sum_{i=1}^k v^i$ for $k \in \{1, \ldots, n\}$.

We now show that in our setting there may not exist an allocation that weakly Lorenz-dominates every other allocation.

Example 7. Suppose there is a single facility with three available units, and there are four agents—namely, agents 1 through 4—such that agent 1 demands all three units from the facility,

[7] While the strategyproofness result of Bogomolnaia and Moulin [7] more generally applies to strategic manipulations from both sides of the market, this is captured by our generalized results in Section 3.2.

while the remaining agents demand a single unit each. Suppose there exists an allocation A that weakly Lorenz-dominates every other feasible allocation. Then, in particular, it must achieve the maximum possible lowest utility. Hence, allocation A must assign agent 1 to the facility with probability 0.5, and assign the remaining agents to the facility simultaneously with the remaining probability 0.5. Thus, the sum of the first three lowest utilities under A is 1.5. However, for the allocation that assigns agents 2 through 4 to the facility with probability 1, the sum of the three lowest utilities is 2, violating our assumption that A weakly Lorenz-dominates every other feasible allocation. Thus, in this case there does not exist any allocation that Lorenz-dominates every other allocation.

In general, the leximin allocation may not be unique, but all leximin allocations are equivalent in a sense formalized in the next result, which is reminiscent of a number of similar results in the literature, e.g., uniqueness of the optimal utility vector when a concave social welfare function (for instance, the Nash social welfare) is maximized over a convex space [2], uniqueness of the nucleolus in cooperative game theory [14], and the "rural hospital theorem" in two-sided matching [33].

Theorem 8. *The utility of an agent is identical under all leximin allocations.*

Proof. Suppose for contradiction that there exist leximin-optimal allocations A and B such that the utilities of some agents do not match in the two allocations. Choose an agent $i \in \arg\min_{i \in N : u_i(A_i) \neq u_i(B_i)} u_i(A_i)$, and break ties by choosing an agent with the smallest $u_i(B_i)$ (further ties can be broken arbitrarily). First, prefix optimality of A (Lemma 5) implies that agent i must be worse off under B, i.e., $u_i(B_i) < u_i(A_i)$. This is because otherwise there would exist an agent $j \in \text{pref}_A(i)$ that is strictly worse off under B. Agent j would satisfy $u_j(B_j) < u_j(A_j) \leq u_i(A_i) < u_i(B_i)$, and thus contradict our choice of agent i. Hence, we have $u_i(B_i) < u_i(A_i)$.

Now, consider the prefix of agent i in B, i.e., $\text{pref}_B(i)$. For every agent $j \in \text{pref}_B(i)$, either agent j has identical utility under A and B (i.e., $u_j(A_j) = u_j(B_j)$), or its utility changes in which case we must have $u_j(A_j) \geq u_i(A_i) > u_i(B_i) \geq u_j(B_j)$, where the first transition holds due to our choice of agent i. Hence, no agent in $\text{pref}_B(i)$ is worse off under A compared to B, and agent i is strictly better off under A compared to B. This violates prefix-optimality of B, which is a contradiction. Hence, the utility of each agent must be identical under all leximin-optimal allocations. □

Crucially, this also implies that all leximin-optimal allocations satisfy an equal number of agents in expectation, and allocate an equal number of units in expectation.

3.2 A General Framework for Leximin

Theorem 3 established that the leximin mechanism satisfies four compelling desiderata in our classroom allocation setting. We observe that the proof of Theorem 3 only uses four characteristics of the classroom allocation setting (which are listed below). That is, the leximin mechanism (Algorithm 1) satisfies proportionality, envy-freeness, Pareto optimality, and group strategyproofness in all domains of fair division and mechanism design without money—with divisible or indivisible (or both types of) resources, and with deterministic or randomized allocations—that satisfy these four requirements.

We briefly describe a general framework in which our result holds. Let N denote the set of agents. There is a set of resources X, which may contain divisible resources, indivisible resources, or both. An allocation A assigns a *disjoint subset* of resources A_i to each agent i.[8] Denote the set of all feasible allocations by $\mathcal{A}$. Note that the use of randomized allocations may or may not be permitted in the domain; it does not affect our result. There is a set $\mathcal{P}$ of possible preferences that the agents may have over possible allocations. Fix a mapping from each preference $P \in P$ to

[8]Obviously, only divisible resources can be split among multiple agents.

a utility function u_P consistent with P, and let $\mathcal{U} = \{u_P | P \in \mathcal{P}\}$ denote the corresponding set of possible utility functions. Then, our four requirements can be formalized as follows.

(1) *Convexity.* The space of feasible allocations must be convex. That is, given two allocations $A, A' \in \mathcal{A}$, and $0 \leq \lambda \leq 1$, it should be possible to construct another feasible allocation $A'' \in \mathcal{A}$ such that $u_i(A''_i) = \lambda \cdot u_i(A_i) + (1 - \lambda) u_i(A'_i)$ for all agents $ti \in N$. This typically holds if randomized allocations are allowed, or if resources are divisible.
(2) *Equality.* The maximum utility achievable by each agent must be identical. Thus, for two agents $i, j \in N$, we require $\max_{A \in \mathcal{A}} u_i(A_i) = \max_{A \in \mathcal{A}} u_j(A_j)$. This property is required for proportionality, and is usually taken care of when translating the ordinal preferences of agents into cardinal utility functions.
(3) *Shifting Allocations.* Given a feasible allocation $A \in \mathcal{A}$ and agents $i, j \in N$, it should be possible to construct another feasible allocation $A' \in \mathcal{A}$ where we take the resources allocated to agent j, and allocate them to agent i. That is, we must have $u_k(A'_k) = u_k(A_k)$ for all agents $k \in N \setminus \{i, j\}$, and $u_i(A'_i) \geq u_i(A_j)$. This property is required for envy-freeness.
(4) *Optimal Utilization.* Under a non-wasteful allocation $A \in \mathcal{A}$, an agent must derive the maximum possible utility from the allocation it receives. That is, we require $u_i(A_i) \geq u(A_i)$ for all possible utility functions $u \in \mathcal{U}$. Lemma 4 proves that this is satisfied in the classroom allocation setting. This assumption is perhaps the most stringent, and is required for both envy-freeness and group strategyproofness.

Many papers study the leximin mechanism and establish (at least a subset of) the properties listed in Theorem 3 in a variety of domains, including resource allocation [5, 7, 18, 23, 28], cake cutting [13], and kidney exchange [34]. It can be checked that these domains satisfy our four requirements, and hence, *Theorem 3 applied to the foregoing framework generalizes results from all of these papers.*

In addition, any general dichotomous preference setting—where each agent "accepts" a subset of feasible allocations for which it has utility 1, and "rejects" the rest for which it has utility 0—is also captured under our general framework; and when agents have ordinal preferences over allocations, we only need to establish one translation to consistent cardinal utilities that satisfies the four requirements above.

Below, we briefly describe one special case of the general framework: fair resource allocation under Leontief preferences [18, 28]. Suppose there are m divisible resources, and each agent i demands them in fixed proportions given by a (normalized) demand vector $\mathbf{d} = (d_{i,1}, \ldots, d_{i,m})$ where $\max_{r \in \{1,\ldots,m\}} d_{i,r} = 1$. Thus, given an allocation $A_i = (A_{i,1}, \ldots, A_{i,m})$ (where $A_{i,r} \in [0, 1]$ denotes the fraction of resource r allocated to agent i), the utility to agent i is given by $u_i(A_i) = \min_{r \in \{1,\ldots,m\}} A_{i,r}/d_{i,r}$. To see that our four requirements are met, note that the space of feasible allocations is convex due to divisibility of resources, every agent can achieve a maximum utility of 1, and shifting allocations is permitted. Finally, a non-wasteful allocation always allocates resources in the demanded proportion. Thus, the utility to agent i is simply $A_{i,r}/d_{i,r}$ (which is identical for all r). Under any other normalized demand vector $d' = (d'_1, \ldots, d'_m)$ with $d'_{r^*} = 1$, the utility achieved would be at most $A_{i,r^*} \leq A_{i,r^*}/d_{i,r^*}$. Hence, the requirement of optimal utilization also holds.

Ghodsi et al. [18] prove that the leximin mechanism satisfies proportionality, envy-freeness, Pareto optimality, and strategyproofness in the foregoing setting, and Parkes et al. [28] establish group strategyproofness. These results now directly follow from Theorem 3. Further, Parkes et al. [28] study the variant where agents only derive utility for integral multiples of their required resource bundle, and show that no deterministic mechanism satisfies all four desiderata. Indeed, in our framework the convexity requirement is violated for deterministic allocations, but it is satisfied for randomized allocations. Hence, the randomized leximin mechanism would still satisfy all four desiderata.

4 QUANTITATIVE EFFICIENCY OF THE LEXIMIN ALLOCATION

Theorem 3 establishes the leximin mechanism as a compelling solution, which simultaneously guarantees fairness, efficiency, and truthfulness. The fairness (proportionality and envy-freeness) and truthfulness guarantees are strong. But the notion of Pareto optimality is a relatively weak, qualitative notion of efficiency.

In our setting, there are two natural *quantitative* metrics of efficiency: the (expected) number of agents whose demands are met, and the (expected) number of total units allocated. Optimizing the former metric is clearly desirable as it represents the social welfare achieved by the mechanism. The latter metric is important when the units being allocated are valuable and scarce (this is clearly the case when the units in question are classrooms). Furthermore, in the classroom allocation setting, the number of units allocated is proportional to the number of students served.

Indeed, in our setting it is not unnatural to consider directly optimizing either metric. In particular, such an optimization would always lead to a Pareto optimal allocation. However, it is easy to observe that directly optimizing either metric fails to achieve one or more of our four desired properties. Recall Example 1, which already showed that maximizing the number of allocated units violates proportionality, envy-freeness, and strategyproofness; the next example deals with the other metric.

Example 9 (Maximizing the number of satisfied agents). Suppose there is a single facility with two available units, and there are four agents, namely, agents 1 through 4. Agents 1 through 3 each demand a single unit from the facility, while agent 4 demands both units. In order to maximize the number of satisfied agents we must allocate a single unit to two of the agents in $\{1, 2, 3\}$, while leaving agent 4 unallocated. It is easy to see that both proportionality (with respect to agent 4) and envy-freeness (with respect to the unallocated agent in $\{1, 2, 3\}$) are violated.

In the above example, proportionality is clearly violated, but it seems that the violation of envy-freeness is the result of tie-breaking. Indeed, as previously mentioned, the utilitarian mechanism [8, 15] that uniformly randomizes over all deterministic allocations maximizing the number of satisfied agents achieves envy-freeness along with strategyproofness. We note that strategyproofness would also hold if ties were broken according to a lexicographic order over the agents (i.e., if ties are broken in favor of allocations satisfying the first agent in the ordering, remaining ties are broken in favor of allocations satisfying the second agent in the ordering, and so on). Here, we provide a short proof of these results for curious readers.

Observation 10. *The mechanism that returns an allocation maximizing the number of satisfied agents and breaks ties according to a lexicographic preference over agents is strategyproof and Pareto optimal. Breaking ties uniformly at random preserves these properties, and achieves envy-freeness.*

Proof. Both the deterministic and the randomized mechanisms are clearly Pareto optimal. We now show strategyproofness of both mechanisms, and envy-freeness of the randomized mechanism. Let A denote the allocation returned by the deterministic mechanism.

Strategyproofness. Suppose agent $i \in N$ is not satisfied under A. Suppose agent i manipulates, which results in allocation A' satisfying agent i. Let k and k' denote the number of agents satisfied in A and A', respectively. Since agent i cannot decrease its demanded number of units, any subset of agents satisfiable after the manipulation is also satisfiable before the manipulation. Hence, $k \geq k'$. However, allocation A does not assign agent i to any facility, and therefore must be feasible after the manipulation. Thus, $k' \geq k$, implying $k = k'$. Finally, note that the subset of agents satisfied by A' was feasible before manipulation, but was not chosen because the subset of agents satisfied under A was better in the lexicographic preference. Since A is a feasible allocation after manipulation, it would still be preferred to A' under the same lexicographic preference, thus establishing a contradiction.

Suppose the mechanism returns an allocation A that uniformly randomizes over all allocations maximizing the number of satisfied agents. Let A' denote the corresponding (uniformly randomizing) allocation when agent i manipulates. If agent i is satisfied with probability 1 under A, then it has no incentive to manipulate. Otherwise, there exists an allocation in the support of A that does not satisfy agent i. Observing that this allocation is feasible after manipulation, and that every subset of agents satisfiable after manipulation is also satisfiable before manipulation, we again get $k = k'$. Moreover, since agent i cannot decrease its demand, the number of allocations in the support of A' in which agent i is satisfied is at most the number of such allocations in A. Since both A and A' uniformly randomize over allocations in their support, it is clear that agent i cannot increase its utility by manipulating.

Envy-Freeness. Consider agents $i, j \in N$. Suppose for contradiction that agent i envies agent j. Let I denote the set of deterministic allocations in the support of A in which agent i is assigned to a facility, while agent j is unassigned. Let J denote the set of deterministic allocations in the support of A in which agent j is assigned to a facility *that is acceptable to agent* i, while agent i is unassigned. Let p_I and p_J denote the probabilities by which A executes an assignment from I and J, respectively. Then, we must have $p_J > p_I$. However, since agent i envies agent j, we must have $d_j \geq d_i$. Thus, taking an allocation from J, and replacing agent j with agent i must form a feasible allocation. Thus, $|I| \geq |J|$. Due to uniform randomization over all allocations in the support, we get $p_I \geq p_J$, which is a contradiction. (Proof of Observation 10) □

While the utilitarian mechanism seems intriguing, recall that in Example 9 the demand of agent 4 was met with zero probability, suggesting that the mechanism is biased against agents with larger demands. Bogomolnaia et al. call this effect the "tyranny of the majority." While such a bias may be acceptable in some settings, in other settings—classroom allocation, in particular—it is problematic. The bias is formally captured by noting that the utilitarian mechanism *violates proportionality.*

The discussion above leads us to a natural question: *How well does the leximin mechanism perform with respect to the two quantitative notions of efficiency, namely, the number of satisfied agents and the number of allocated units?* We are interested in the worst case over problem instances, but since the leximin mechanism is randomized, we can consider the performance under the worst deterministic allocation in the support of the randomized leximin allocation, and the performance in expectation. Unsurprisingly, the worst allocation in the support can be simultaneously bad in terms of both metrics; in the following example, both metrics achieve arbitrarily low fractions of their respective optimums.

Example 11 (Efficiency of allocations in the support of the leximin allocation). Suppose there are $k + 4$ agents and two facilities. The capacities of the two facilities are $c_1 = k$ and $c_2 = k^2$. The preferences of the agents are as follows.

$$(d_i, F_i) = \begin{cases} (1, \{1\}) & \text{if } i \in \{1, \ldots, k\}, \\ (k, \{1\}) & \text{if } i = k+1 \text{ or } k+2, \\ (1, \{2\}) & \text{if } i = k+3, \\ (k^2, \{2\}) & \text{if } i = k+4. \end{cases}$$

Clearly, a maximum of $k + 1$ agents can be satisfied, and a maximum of $k + k^2$ units can be allocated. It is easy to check that under the leximin allocation, agents 1 through $k + 2$ should be assigned to facility 1 with probability $1/3$ each, while agents $k+3$ and $k+4$ should be assigned to facility 2 with probability $1/2$ each. However, this implies that the support of the leximin allocation must include a deterministic allocation in which agent $k+3$ is assigned to facility 2 while one of agents $k+1$ and $k+2$ is assigned to facility 1 (and the remaining agents are unassigned). In this allocation, the number of agents satisfied is a mere $2/(k+1)$ fraction of the optimum, and the number of units allocated is also a mere $(k+1)/(k+k^2) = 1/k$ fraction of the optimum. Thus, both approximation ratios converge to 0 as k goes to infinity.

Let us therefore consider the worst-case (over instances) performance of the leximin mechanism in expectation (over the randomness of the mechanism). We can show that approximating (in expectation) the maximum number of satisfied agents is directly at odds with proportionality—recall that this is exactly the property that the utilitarian mechanism [8, 15] fails to achieve.

Example 12 (Proportionality and maximizing the number of satisfied agents). Suppose there is a single facility with k units available, and there are $k+k^2$ agents, k of which require 1 unit each while the other k^2 agents require all k units each. Any proportional mechanism must allocate the k units to each of the k^2 agents demanding them with probability at least $1/(k+k^2)$. Hence, such a mechanism satisfies a single agent with probability at least $k^2/(k+k^2)$, and at most k agents with the remaining probability. Therefore, the expected number of satisfied agents is at most $k^2/(k+k^2) + k \cdot k/(k+k^2) \leq 2$. However, a maximum of k agents could be satisfied simultaneously. Hence, any proportional mechanism (including the leximin mechanism) achieves an approximation ratio of at most $2/k$ for the number of satisfied agents. This ratio goes to 0 as k goes to infinity.

In contrast, we make the following conjecture for the expected number of *units* allocated by the leximin mechanism.

Conjecture 13. *The expected number of units allocated by the leximin mechanism 2-approximates the maximum number of units that can be allocated simultaneously by any non-wasteful allocation (in the worst case over instances).*

The conjecture is based on millions of randomly generated instances. In all of these instances, the leximin mechanism allocated, in expectation, at least half of the optimal number of units. While the conjecture is still open, we are able to prove a slightly weaker 4-approximation result.

Theorem 14. *The expected number of units allocated by the leximin mechanism 4-approximates the maximum number of units that can be allocated simultaneously by any non-wasteful allocation (in the worst case over instances).*

Proof. Let us first prove a 2-approximation in the case of a single facility to gain some intuition. Let c denote the capacity of the facility, and D denote the maximum number of units allocated by a non-wasteful allocation. If all the deterministic assignments in the support of the leximin allocation allocate at least $D/2$ units, then the result follows trivially. Suppose a deterministic assignment allocates $t < D/2 \leq c/2$ units to agents in $S \subseteq N$, and is realized with probability p. Hence, it is clear that $N \setminus S \neq \emptyset$. Due to Pareto optimality of the leximin allocation, an allocation that does not assign any agent in $N \setminus S$ to the facility must assign all agents in S to the facility. That is, there is a unique such allocation, which is realized with probability p. Further, due to the nature of the leximin allocation, every agent in $N \setminus S$ must also be assigned to the facility with probability at least p, implying that $p \leq 1/2$. Thus, with probability $p \leq 1/2$ the mechanism allocates t units, and with the remaining probability $1-p$ the mechanism assigns at least one agent in $N \setminus S$ to the facility, thus allocating more than $c-t$ units (this is because every agent in $N \setminus S$ must have demand greater than $c - t$ due to Pareto optimality of the leximin allocation). Hence, the expected number of units allocated is at least $t \cdot 1/2 + (c-t) \cdot 1/2 = c/2 \geq D/2$.

However, generalizing this proof to achieve a "per facility" constant approximation is difficult. Instead, our proof below works in three steps.

(1) We fix an arbitrary (deterministic) allocation A^* that maximizes the number of units allocated.
(2) Next, after adding certain "virtual allocated units" to each facility (derived based on A^*), the expected number of units allocated by the leximin mechanism 2-approximates the number of units allocated under A^* *on each facility individually*.
(3) Finally, we show that the expected number of virtual units added overall is no more than the expected number of units allocated by the leximin mechanism, thus establishing the 4-approximation result.

Let A^* denote an arbitrary deterministic allocation that maximizes the number of units allocated. For a facility $f \in M$, let $Z(f) = \{i \in N | A_i^* = f\}$ denote the set of agents assigned to facility f under A^*. Let L denote the leximin allocation, which executes deterministic allocation L^k with probability p_k for $k \in \{1, \ldots, T\}$. We are now ready for our main lemma. For a facility $f \in M$, the number of "virtual units" we add is the expected number of units allocated by the leximin mechanism to the agents in $Z(f)$ (at any facility). We show that the expected number of units allocated by the leximin mechanism at facility f and the number of virtual units for facility f together 2-approximate the number of units allocated by A^* at facility f, for each $f \in M$. □

Lemma 15. *For a facility $f \in M$ we have*

$$\sum_{k=1}^{T} p_k \left(\sum_{i \in N : L_i^k = f} d_i + \sum_{i \in Z(f) : L_i^k \neq 0} d_i \right) \geq \frac{1}{2} \sum_{i \in Z(f)} d_i .$$

Proof. Let us consider two cases.

Case 1: For every $i \in Z(f)$, $d_i \leq c_f/2$. In this case we can show that

$$\sum_{i \in N : L_i^k = f} d_i + \sum_{i \in Z(f) : L_i^k \neq 0} d_i \geq \frac{1}{2} \sum_{i \in Z(f)} d_i \tag{1}$$

for each $k \in \{1, \ldots, T\}$. If $L_i^k \neq 0$ for every $i \in Z(f)$, then the second term in the left-hand side (LHS) of Equation (1) is at least $\sum_{i \in Z(f)} d_i$. Otherwise, let $L^k(\gamma) = 0$ for some $\gamma \in Z(f)$. By the Pareto optimality of L^k, we know that the demand of agent γ must be greater than the number of unallocated units at facility f in L^k, i.e.,

$$d_\gamma > c_f - \sum_{i \in N : L_i^k = f} d_i .$$

Using $d_\gamma < c_f/2$, we get that the first term in the LHS of Equation (1) is greater than the right-hand side (RHS). Hence, in either case Equation (1) holds.

Case 2: There exists an agent $\gamma \in Z(f)$ such that $d_\gamma > c_f/2$. Let us define two sets.

(1) $I = \{k \in \{1, \ldots, T\} \mid L_\gamma^k \neq 0\}$.
(2) $J = \{k \in \{1, \ldots, T\} \mid L_\gamma^k = 0 \text{ and } \sum_{i \in N : L_i^k = f} d_i < c_f/2\}$.

Furthermore, let $p_I = \sum_{k \in I} p_k$ and $p_J = \sum_{k \in J} p_k$. We claim that $p_I \geq p_J$. Note that p_I is precisely the probability that agent γ is satisfied under the leximin allocation.

Suppose for contradiction that $p_I < p_J$. Take some $\ell \in J$, and let $W = \{i \in N \mid L_i^\ell = f\}$. From the definition of J, we know that each agent $i \in W$ must satisfy $d_i < c_f/2$. Further, for each $k \in J$ facility f has more than $c_f/2$ units unallocated in L^k. Hence, by the Pareto optimality of the leximin allocation, every agent in W must be assigned to some facility in L^k for every $k \in J$. Importantly, this implies that every agent in W has probability at least $p_J > p_I$ of being assigned to a facility under the leximin allocation.

Now, fix a small $\epsilon > 0$, and consider a new randomized allocation $\tilde{L}$ that executes deterministic allocations $L^1, \ldots, L^{k-1}, L^k, L^{k+1}, \ldots, L^T$, and L^{T+1} with probabilities $p_1, \ldots, p_{k-1}, (1 - \epsilon)p_k, p_{k+1}, \ldots, p_T$, and ϵp_k, respectively, where

$$L_i^{T+1} = \begin{cases} L_i^k & \text{if } L_i^k \neq f \\ 0 & \text{if } L_i^k = f \text{ and } i \neq \gamma \\ 1 & \text{otherwise.} \end{cases}$$

Note that f must be an acceptable facility to agent γ because $\gamma \in Z(f)$. Hence, allocation L_i^{T+1} respects the preferences of the agents. It is easy to check that the capacity constraint at each facility (including facility f) is also respected. Essentially, we replace all the agents assigned at facility f in L^k by a single agent γ. For a sufficiently small $\epsilon > 0$, we can see that

(1) agent γ has a strictly higher probability of being assigned to a facility under $\tilde{L}$ than under L (under L, it is assigned to a facility with probability exactly p_I);
(2) an agent $i \neq \gamma$ that is assigned to a facility with probability $p \leq p_I$ (thus, from the above argument $L_i^k \neq f$) has the same probability of being assigned to a facility under $\tilde{L}$ as under L;
(3) all the remaining agents were assigned to a facility with probability strictly more than p_I under L, and their probabilities remain strictly greater than p_I under $\tilde{L}$.

However, this contradicts the fact that L is a leximin-optimal allocation. This is essentially a consequence of the prefix optimality of L (Lemma 5). Hence, we have $p_I \geq p_J$, as claimed.

With this claim in hand, we can show the required inequality. Let us consider the sum in the LHS.

$$\sum_{k=1}^{T} p_k \left(\sum_{i \in N : L_i^k = f} d_i + \sum_{i \in Z(f) : L_i^k \neq 0} d_i \right).$$

We break the summation over $k \in I$, $k \in J$, and $k \in \{1, \ldots, T\} \setminus (I \cup J)$. For each $k \in I$, we have $L_\gamma^k \neq 0$. Hence, the term inside the brackets is at least d_γ. For each $k \in J$, we have $L_\gamma^k = 0$. Hence, the term inside the brackets, which is no less than the number of units allocated at facility f in L^k, must be at least $c_f - d_\gamma$. Finally, from definitions of I and J, it follows that the term inside the brackets is at least $c_f/2$ for every $k \in \{1, \ldots, T\} \setminus (I \cup J)$. Hence, we have that the LHS is at least

$$\begin{aligned} &\sum_{k \in I} p_k \cdot d_\gamma + \sum_{k \in J} p_k \cdot (c_f - d_\gamma) + \sum_{k \in \{1, \ldots, T\} \setminus (I \cup J)} p_k \cdot \frac{c_f}{2} \\ &= p_I \cdot d_\gamma + p_J \cdot (c_f - d_\gamma) + (1 - p_I - p_J) \cdot \frac{c_f}{2} \\ &= (p_I - p_J) \cdot d_\gamma + (1 - p_I + p_J) \cdot \frac{c_f}{2} \\ &\geq (p_I - p_J) \cdot \frac{c_f}{2} + (1 - p_I + p_J) \cdot \frac{c_f}{2} = \frac{c_f}{2} \geq \frac{1}{2} \sum_{i \in Z(f)} d_i, \end{aligned}$$

where the third transition holds because $p_I \geq p_J$ and $d_Y \geq c_f/2$. Thus, we have proved that the lemma holds in both the cases we considered. (Proof of Lemma 15) □

Lemma 15 holds for every facility individually. Summing over all facilities, we get

$$\sum_{f \in M} \sum_{k=1}^{T} p_k \left(\sum_{i \in N : L_i^k = f} d_i + \sum_{i \in Z(f) : L_i^k \neq 0} d_i \right) \geq \frac{1}{2} \sum_{f \in M} \sum_{i \in Z(f)} d_i. \tag{2}$$

In Equation (2), we have

$$\begin{aligned} \text{LHS} &= \sum_{k=1}^{T} p_k \cdot \left(\sum_{f \in M} \sum_{i \in N : L_i^k = f} d_i + \sum_{f \in M} \sum_{i \in Z(f) : L_i^k \neq 0} d_i \right) \\ &= \sum_{k=1}^{T} p_k \cdot \left(\sum_{i \in N : L_i^k \neq 0} d_i + \sum_{i \in N : A_i^* \neq 0, L_i^k \neq 0} d_i \right) \\ &\leq 2 \cdot \sum_{k=1}^{T} p_k \left(\sum_{i \in N : L_i^k \neq 0} d_i \right), \end{aligned}$$

$$\text{RHS} = \frac{1}{2} \sum_{f \in M} \sum_{i \in Z(f)} d_i = \frac{1}{2} \sum_{i \in N : A_i^* \neq 0} d_i.$$

Note that *LHS* is at most twice the expected number of units allocated by the leximin mechanism, and *RHS* is half the number of units allocated by A^*. Hence, the expected number of units allocated by the leximin mechanism 4-approximates the maximum number of units allocated by a non-wasteful allocation. □

While we strongly believe that the approximation ratio of Theorem 14 can be improved from 4 to 2, it can easily be seen that a proportional or envy-free mechanism (including the leximin mechanism) cannot achieve an approximation ratio better than 2. Consider the case of a single facility with $2k$ units, and $k+1$ agents, one of which requires all $2k$ units while the rest require $k + 1$ units each. Clearly any proportional or envy-free mechanism must assign each agent demanding $k+1$ units alone to the facility with probability at least $1/(k+1)$. Hence, the expected number of allocated units cannot be more than $(k+1) \cdot k/(k+1) + 2k \cdot 1/(k+1) \leq k+2$, while a maximum of $2k$ units can be allocated simultaneously. This lower bound on the approximation ratio tends to 2 as k tends to infinity.

5 COMPLEXITY AND IMPLEMENTATION

Recall that our classroom allocation setting is a generalization of the classic setting of random assignment under dichotomous preferences studied by Bogomolnaia and Moulin [7] (which can be viewed in our model as restricting agents to have unit demands and facilities to have unit capacities). In the classic setting, leximin allocations can be computed in polynomial time by leveraging the Birkhoff von-Neumann theorem [4, 36].

In contrast, an immediate reduction from PARTITION shows that computing the leximin allocation is $\mathcal{NP}$-hard in our generalized setting. Indeed, consider an instance of PARTITION: given a set S of n integers that sum to $2T$ for $T \in \mathbb{N}$, one needs to decide if there exists a subset $S' \subseteq S$ whose elements sum to T. Construct an instance of our problem in which a single facility has T available units and there are n agents whose demands correspond to the elements of S. Then, the leximin allocation would assign each agent to the facility with probability at least 1/2 if and only if there exists a partition of S.

The standard approach to computing the leximin allocation (see, e.g., [25]) is to successively solve linear programs (LPs) in order to maximize the lowest utility, subject to that maximize the second lowest utility, and so on. While previous work focused on establishing polynomial running time of this approach in various domains, in our domain this task is $\mathcal{NP}$-complete. Hence, in the remainder of the section, we focus on designing optimized heuristics for computing the leximin allocation in the classroom allocation setting. We use a variable p_i to denote the probability that agent i is satisfied, for every $i \in N$. In a naïve implementation, we can include a variable x_A for every possible deterministic assignment $A \in \mathcal{A}$ that represents the probability of executing A, and write $p_i = \sum_{A \in \mathcal{A}: A_i \neq 0} x_A$. However, the number of feasible deterministic allocations can be roughly $(m+1)^n$, which makes the LPs extremely large even for moderately large values of m and n.

Crucially, note that we only care about whether a given agent is satisfied in a deterministic allocation, and not about the facility to which the agent is assigned. In other words, two deterministic allocations that satisfy identical subsets of agents are, in some sense, equivalent. This is due to the dichotomous nature of the preferences of agents over facilities. This observation leads us to our first algorithm, presented as Algorithm LeximinPrimal, which works as follows. First, we compute the collection of "feasible subsets" of agents, i.e., subsets of agents that can be satisfied simultaneously. Let $\mathcal{S} = \{S \subseteq N \mid \exists A \in \mathcal{A} \text{ s.t. } \forall i \in S, A_i \neq 0\}$. Checking feasibility of a given subset of agents S can be encoded as an integer linear program (ILP), presented as FeasibilityILP in the algorithm, which checks if agents in S can be assigned to one of their acceptable facilities while respecting the capacity constraints. Note that a feasible solution to FeasibilityILP also provides an assignment A_S that satisfies S.

Finally, we form an LP, which we call PrimalLP, in which variable x_S denotes the probability by which $S \subseteq N$ is satisfied, and express the individual agent utilities as $p_i = \sum_{S \subseteq N: i \in S} x_S$ for $i \in N$. The algorithm maintains a set of agents R whose utilities in the leximin allocation it has not yet found, and stores the utility of each agent $i \in N \setminus R$ as p_i^*. In each iteration, the algorithm maximizes the (next) minimum utility of agents in R while keeping the utilities of agents in $N \setminus R$ intact, stores the utilities of agents that have the next minimum utility, and removes them from R.

The algorithm clearly terminates because any optimal solution to PrimalLP must set $p_i = M$ for at least one $i \in R$. Hence, $|R|$ decreases by at least 1 in every iteration. Further, if M is the optimal objective value of PrimalLP, then an observation from the convex optimization literature states that there must exist at least one $j \in R$ that has utility M in all optimal solutions to PrimalLP, and in particular, in the actual leximin allocation too.[9] Our use of a strictly complementary solution to PrimalLP ensures that we have $p_j = M$ only if it holds in all optimal solutions.[10] Thus, Algorithm LeximinPrimal always makes "safe" choices, and correctly returns a leximin allocation. Finally, note that the values of p_i^* from one iteration are used to compute p_i^* in the next iteration. While this may lead to an exponential blowup in the length of their binary representation, it does not affect the running time of our algorithm due to a result by Tardos [35].[11] Interestingly, note that the choices made by the algorithm do not affect agent utilities in the returned leximin allocation due to Theorem 8.

[9] If for every $j \in R$ there exists a solution to PrimalLP with $p_j > M$, a positive convex combination of these solutions would be a feasible solution with a strictly greater objective value, which is a contradiction.

[10] Strictly complementary solutions can be found by using any interior point method based on the central trajectory [16], by using a trick due to Freund, Roundy, and Todd [17] which requires solving a single LP using any off-the-shelf solver, or by solving one LP for each $i \in R$ to check if p_i can be made greater than M in some optimal solution to PrimalLP.

[11] This result shows that the running time of an interior point method is independent of the bit length of values on the right-hand side of an LP, which is where the p_i^* are used in PrimalLP.

ALGORITHM 2: LeximinPrimal

Data: Demands $\{(d_i, F_i)\}_{i \in N}$, Capacities $\{c_j\}_{j \in M}$
Result: The Leximin Allocation A
Solve FeasibilityILP for each $S \subseteq N$, and let $\mathcal{S} \leftarrow$ the set of maximal feasible subsets of N;
For each $S \in \mathcal{S}$, $A_S \leftarrow$ the assignment returned by FeasibilityILP on S;
$R = N$;
$p_i^* = 0, \forall i \in N$;
do
    $(M, \{p_i\}_{i \in R}, \{x_S\}_{S \in \mathcal{S}}) \leftarrow$ Strictly complementary solution to PrimalLP in the box below;
    $p_i^* = M, \forall i \in R : p_i = M$;
    $R = R \setminus \{i \in N | p_i = M\}$;
    if $R = \emptyset$ **then**
        return the randomized allocation where A_S is executed with probability x_S for each $S \in \mathcal{S}$;
    end
while $R \neq \emptyset$;

PrimalLP :
Maximize M
subject to
$p_i \geq M, \forall i \in R$
$p_i = p_i^*, \forall i \in N \setminus R$
$p_i = \sum_{S \in \mathcal{S}, i \in S} x_S, \forall i \in N$
$\sum_{S \in \mathcal{S}} x_S = 1$
$x_S \geq 0, \forall S \in \mathcal{S}$

FeasibilityILP :
$\sum_{f \in F_i} y_{i,f} \geq 1, \forall i \in S$
$\sum_{i \in S : f \in F_i} d_i \cdot y_{i,f} \leq c_f, \forall f \in M$
$y_{i,f} \in \{0,1\}, \forall i \in S, f \in F_i$

We employ two further optimizations to reduce the running time of LeximinPrimal: (i) solving FeasibilityILP on different subsets of agents in the decreasing order of their sizes, and only solving it for $S \subseteq N$ if none of its strict supersets are already found to be feasible, and (ii) only using *maximal* feasible subsets in $\mathcal{S}$ because Pareto optimality prevents the leximin allocation from using any non-maximal subset.

Next, we present another algorithm that, instead of solving PrimalLP, solves its dual. This is presented as Algorithm LeximinDual. Note that PrimalLP has polynomially many constraints and exponentially many variables. Correspondingly, its dual (DualLP) has polynomially many variables and exponentially many constraints (in particular, one constraint for each $S \in \mathcal{S}$). We can identify the tight primal constraints ($p_i = M$ for $i \in R$) by simply checking if the corresponding dual variable is strictly positive ($\alpha_i > 0$) due to the strict complementary slackness conditions. We solve DualLP using the Ellipsoid algorithm [21], which makes polynomially many calls to an "oracle" for finding a violated constraint (if one exists) given any values of the variables. Crucially, we observe that finding $S \in \mathcal{S}$ that corresponds to the most violated constraint can be encoded as an ILP, presented along with the algorithm. We use $\hat{\mathcal{S}}$ to denote the polynomial-size collection of subsets of agents on which the oracle is called by the Ellipsoid algorithm. There are three special advantages of the oracle:

(1) Since the oracle includes feasibility constraints, we can avoid the initial (computationally expensive) stage of LeximinPrimal solving FeasibilityILP for 2^n subsets of agents, and instead solve only polynomially many ILPs for subsets in $\hat{\mathcal{S}}$.
(2) Since LeximinDual makes only polynomially many calls to the oracle, the overall space complexity is polynomial. In particular, the returned leximin allocation randomizes over

ALGORITHM 3: LeximinDual

Data: Demands $\{(d_i, F_i)\}_{i \in N}$, Capacities $\{c_j\}_{j \in M}$
Result: The Leximin Allocation A
$R = N$;
$p_i^* = 0, \forall i \in N$;
do
  $(M, \{\alpha_i\}_{i \in R}) \leftarrow$ Strictly complementary solution to DualLP in the box below;
  $p_i^* = M, \forall i \in R : \alpha_i > 0$;
  $R = R \setminus \{i \in N | p_i = M\}$;
while $R \neq \emptyset$;
$\hat{\mathcal{S}} \leftarrow \{S \subseteq N|$ oracle of DualLP was called on S in the last iteration of the loop$\}$;
For each $S \in \hat{\mathcal{S}}$, $A_S \leftarrow$ the assignment returned by the oracle when it was called on S;
$\{x_S\}_{S \in \hat{\mathcal{S}}} \leftarrow$ Solution to FinalLP in the box below;
return the randomized allocation where A_S is executed with probability x_S for each $S \in \hat{\mathcal{S}}$;

DualLP :

Min. $M = \delta - \sum_{i \in N \setminus R} p_i^* \cdot \beta_i$
subject to
$\sum_{i \in R} \alpha_i = 1$
$-\alpha_i - \gamma_i = 0, \forall i \in R$
$-\beta_i - \gamma_i = 0, \forall i \in N \setminus R$
$\delta + \sum_{i \in S} \gamma_i \geq 0, \forall S \in \mathcal{S}$
$\alpha_i \geq 0, \forall i \in R$

Oracle for DualLP :

Max. $\sum_{i \in N} \gamma_i \cdot \left(\sum_{j \in F_i} y_{i,j} \right)$
subject to the constraints of FeasibilityILP

FinalLP :

Find a feasible solution to
$p_i = p_i^*, \forall i \in N$
$p_i = \sum_{S \in \hat{\mathcal{S}}, i \in S} x_S, \forall i \in N$
$\sum_{S \in \hat{\mathcal{S}}} x_S = 1$
$x_S \geq 0, \forall S \in \hat{\mathcal{S}}$

polynomially many subsets of agents (i.e., it is *sparse*), making it more feasible to store and implement the allocation in practice.

(3) In special cases such as the case of unit demands and capacities (i.e., the classic random assignment setting studied by Bogomolnaia and Moulin [7]), the oracle can be encoded as a polynomial-size LP by leveraging the Birkhoff von-Neumann theorem [4, 36], which would automatically make the overall running time of LeximinDual polynomial.

In the next section, we show that LeximinDual is actually drastically superior to LeximinPrimal in terms of running time.

6 EXPERIMENTS

Our goal in this section is to empirically compare algorithms LeximinPrimal and LeximinDual, as well as evaluate the performance of the leximin allocation in terms of the number of satisfied agents and the number of allocated units.

In our experiments, we vary the number of agents n from 5 to 300.[12] Note that the largest school district in the U.S. (by the number of charter schools) is the LAUSD which has 241 charter schools.[13] We observe that in practice the number of facilities varies from about $5n$ (for LAUSD) to about $20n$ (for PUSD). Thus, we select m uniformly at random from the interval $[5n, 20n]$. Next,

[12] We use $n = 5, 10, 15$ for LeximinPrimal as it fails to run beyond that, and evaluate LeximinDual further on $n = 50, 100, 150, 200, 250, 300$.
[13] Refer to http://goo.gl/Bu0pz9 and http://goo.gl/ILJupc.

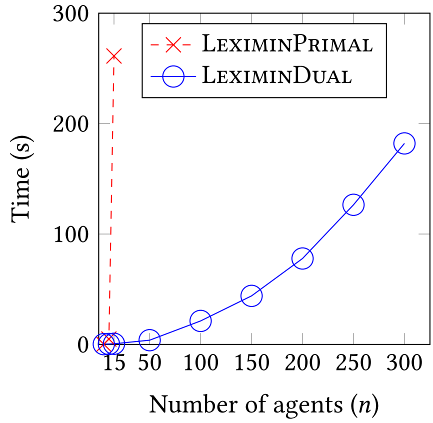


Fig. 2. Running time of LeximinPrimal and LeximinDual.

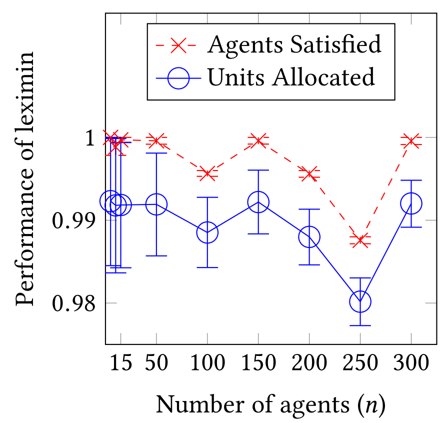


Fig. 3. Performance of the leximin allocation as a fraction of the optimum.

we fit Poisson distributions to the real-world demands and capacities data from PUSD, and use them to generate demands and capacities in our experiments. For the dichotomous preferences of agents over facilities, we observe that in the PUSD data certain facilities were inherently more desirable than others, and were accordingly accepted by many charter schools. We thus generate a "quality parameter" for each facility in $[0, 1]$ from the beta distribution with both parameters equal to 5, and have each agent accept the facilities (which have sufficient capacity to meet its demand) with probabilities proportional to their qualities. For each value of n, the values in all our graphs are averaged over 500 simulations. We use MATLAB to obtain strictly complementary solutions to linear programs, and CPLEX to solve integer linear programs. Our experiments are performed on an Intel PC with dual core, 3.10GHz processors, and 8GB RAM.

Figure 2 compares the running time of algorithms LeximinPrimal and LeximinDual. Note that the running time of LeximinPrimal increases extremely quickly as n grows, making it infeasible to run the algorithm beyond $n = 15$. In contrast, LeximinDual solves instances with $n = 300$ (recall that this is larger than any real-world instance) in just a little over 3 minutes. This is a direct result of the fact that LeximinDual ends up solving less than 1% of the ILPs solved by LeximinPrimal, and solving ILPs is the bottleneck in both algorithms. Another interesting fact is that the number of times the loop in LeximinDual (or in LeximinPrimal) runs is equal to the number of distinct utility values in the leximin solution, because all agents with identical utilities are removed in a single iteration. The number of iterations required is less than three on average in our simulations. We also remark that even if the Proposition 39 process scaled to the state level, California has approximately 1130 charter schools overall,[13] and LeximinDual can also solve such huge instances in less than 2 hours (this result is averaged over 10 simulations).

Next, in Figure 3 we show the ratios of the expected number of agents satisfied and the expected number of units allocated by the leximin mechanism to the maximum possible values of the respective metrics. Remarkably, both ratios stay above a whopping 0.98 on average, which is significantly better than the upper bounds on the worst-case (over possible instances) performance of the leximin mechanism (almost 0 for the expected number of agents satisfied and 1/2 for the expected number of units allocated). The error bars show confidence intervals for the performance of the deterministic allocations in the support of the leximin allocation. Specifically, we remove the best (the worst, respectively) deterministic allocations with an aggregate probability of at most 0.1 from the support, and then measure the best (the worst, respectively) performance of any deterministic allocation in the support. A final remark is that the size of the support of the leximin allocation is less than 8 on average in our simulations. A randomization over at most eight deterministic allocations can easily be stored and implemented in practice, which further supports the practicability of the leximin mechanism.

7 EPILOGUE AND DISCUSSION OF PRACTICAL ASPECTS

In January 2015, PUSD asked charter schools to formally report dichotomous preferences, in addition to the free-text preferences submitted through the usual request form. The plan was to evaluate our approach by comparing its output on the collected explicit dichotomous preferences against human-generated allocations based on the free-text preferences. Despite the promising outlook, sadly, in April 2015 the collaboration was terminated by PUSD, for reasons unknown to us. Nonetheless, we were informed that this initiative helped PUSD build a good rapport with local charter schools.

Meanwhile, Mr. Mopatis put us in touch with representatives of the LAUSD), the largest school district in California with 274 charter schools and over 900 public schools, which is perfect for highlighting the advantages of our automated approach over human-generated allocations. We are planning to reach out to additional school districts, and, while the process of deploying our algorithm is slower and more complicated than we initially expected, we are hopeful that our approach will be evaluated in the future.

On a practical level, the simplicity of the leximin mechanism, and the intuitiveness of the properties of proportionality, envy-freeness, Pareto optimality, and strategyproofness, have made the approach more likely to be adopted. On the other hand, the use of randomization, though absolutely necessary in order to guarantee fairness in allocating indivisible goods such as classrooms, has been a somewhat harder sell. Ironically, this seems to be the result of presenting the mechanism as a "lottery," which makes it easier to comprehend on the one hand, but on the other hand raises negative connotations and legal objections—even though many charter schools use a (straightforward) lottery system to admit students. In terms of lessons learned, it actually seems better to use more technical terms in this context.

In conclusion, redesigning the way California's school districts allocate classrooms to charter schools is a major project with the potential for societal impact. This article presents a detailed technical approach, but deployment of this approach is still in its infancy; we hope to continue working with school districts for years to come.

REFERENCES

[1] A. Abdulkadiroğlu, T. Sönmez, and M. U. Ünver. 2004. Room assignment-rent division: A market approach. *Social Choice and Welfare* 22, 3 (2004), 515–538.

[2] N. Anari, T. Mai, S. O. Gharan, and V. V. Vazirani. 2016. Nash social welfare for indivisible items under separable, piecewise-linear concave utilities. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'18)*, 2274–2290.

[3] H. Aziz. 2014. Random assignment with multi-unit demands. *arXiv:1401.7700.*

[4] G. Birkhoff. 1946. Three observations on linear algebra. *Universidad Nacional de Tucumán, Revista A* 5 (1946), 147–151.

[5] O. Bochet, R. Ilkılıç, H. Moulin, and J. Sethuraman. 2012. Balancing supply and demand under bilateral constraints. *Theoretical Economics* 7, 3 (2012), 395–423.

[6] A. Bogomolnaia and H. Moulin. 2001. A new solution to the random assignment problem. *Journal of Economic Theory* 100 (2001), 295–328.

[7] A. Bogomolnaia and H. Moulin. 2004. Random matching under dichotomous preferences. *Econometrica* 72 (2004), 257–279.

[8] A. Bogomolnaia, H. Moulin, and R. Stong. 2005. Collective choice under dichotomous preferences. *Journal of Economic Theory* 122, 2 (2005), 165–184.

[9] S. J. Brams and A. D. Taylor. 1996. *Fair Division: From Cake-Cutting to Dispute Resolution.* Cambridge University Press.

[10] E. Budish. 2011. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy* 119, 6 (2011), 1061–1103.
[11] E. Budish, Y.-K. Che, F. Kojima, and P. Milgrom. 2013. Designing random allocation mechanisms: Theory and applications. *American Economic Review* 103, 2 (2013), 585–623.
[12] Y. K. Che and F. Kojima. 2010. Asymptotic equivalence of probabilistic serial and random priority mechanisms. *Econometrica* 78, 5 (2010), 1625–1672.
[13] Y. Chen, J. K. Lai, D. C. Parkes, and A. D. Procaccia. 2013. Truth, justice, and cake cutting. *Games and Economic Behavior* 77 (2013), 284–297.
[14] T. S. H. Driessen. 1988. *Cooperative Games, Solutions and Applications*. Springer.
[15] G. Freitas. 2010. Combinatorial Assignment under Dichotomous Preferences. Manuscript.
[16] R. M. Freund and S. Mizuno. 2000. *Interior Point Methods: Current Status and Future Directions*. Springer.
[17] R. M. Freund, R. Roundy, and M. Todd. 1985. *Identifying the Set of Always-Active Constraints in a System of Linear Inequalities by a Single Linear Program*. Technical Report 1674-85. Massachusetts Institute of Technology (MIT), Sloan School of Management.
[18] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, and I. Stoica. 2011. Dominant resource fairness: Fair allocation of multiple resource types. In *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation (NSDI'11)*. 24–37.
[19] J. Goldman and A. D. Procaccia. 2014. Spliddit: Unleashing fair division algorithms. *SIGecom Exchanges* 13, 2 (2014), 41–46.
[20] A. Hylland and R. Zeckhauser. 1979. The efficient allocation of individuals to positions. *The Journal of Political Economy* 87, 2 (1979), 293–314.
[21] L. Khachiyan. 1979. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady* 20 (1979), 191–194.
[22] F. Kojima. 2009. Random assignment of multiple indivisible objects. *Mathematical Social Sciences* 57, 1 (2009), 134–142.
[23] W. Li, X. Zhang, and X. Zhang. 2014. Multi-Resource Fair Allocation with Bounded Number of Tasks in Cloud Computing Systems. arXiv:1410.1255.
[24] H. Moulin. 2003. *Fair Division and Collective Welfare*. MIT Press.
[25] Dritan Nace and James B. Orlin. 2007. Lexicographically minimum and maximum load linear programming problems. *Operations Research* 55, 1 (2007), 182–187.
[26] A. Othman, C. H. Papadimitriou, and A. Rubinstein. 2014. The complexity of fairness through equilibrium. In *Proceedings of the 15th ACM Conference on Electronic Commerce (EC'14)*. 209–226.
[27] A. Othman, T. Sandholm, and E. Budish. 2010. Finding approximate competitive equilibria: Efficient and fair course allocation. In *Proceedings of the 9th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'10)*. 873–880.
[28] D. C. Parkes, A. D. Procaccia, and N. Shah. 2015. Beyond dominant resource fairness: Extensions, limitations, and indivisibilities. *ACM Transactions on Economics and Computation* 3, 1 (2015).
[29] E. Pazner and D. Schmeidler. 1978. Egalitarian equivalent allocations: A new concept of economic equity. *Quarterly Journal of Economics* 92, 4 (1978), 671–687.
[30] A. D. Procaccia. 2013. Cake cutting: Not just child's play. *Communications of the ACM* 56, 7 (2013), 78–87.
[31] M. Pycia. 2011. Assignment with Multiple-Unit Demands and Responsive Preferences. Manuscript.
[32] J. M. Robertson and W. A. Webb. 1998. *Cake Cutting Algorithms: Be Fair If You Can*. A. K. Peters.
[33] A. E. Roth. 1986. On the allocation of residents to rural hospitals: A general property of two-sided matching markets. *Econometrica* (1986), 425–427.
[34] A. E. Roth, T. Sönmez, and M. U. Ünver. 2005. Pairwise kidney exchange. *Journal of Economic Theory* 125 (2005), 151–188.
[35] E. Tardos. 1986. A strongly polynomial algorithm to solve combinatorial linear programs. *Operations Research* 34, 2 (1986), 250–256.
[36] J. von Neumann. 1953. A certain zero-sum two-person game equivalent to the optimal assignment problem. In *Contributions to the Theory of Games*, W. Kuhn and A. W. Tucker (Eds.). Vol. 2. 5–12.

Received December 2015; revised July 2017; accepted May 2018