Welfare Maximization and Truthfulness in Mechanism Design with Ordinal Preferences*

Deeparnab Chakrabarty
Microsoft Research
9 Lavelle Road, Bangalore 560001, India.
dechakr@microsoft.com

Chaitanya Swamy†
Dept. of Comb. and Opt., Univ. Waterloo
200 University Avenue, Waterloo, ON N2L 3G1
cswamy@math.uwaterloo.ca

ABSTRACT

In this paper, we study mechanism design problems in the *ordinal setting* wherein the preferences of agents are described by orderings over outcomes, as opposed to specific numerical values associated with them. This setting is relevant when agents can compare outcomes, but aren't able to evaluate precise utilities for them. Such a situation arises in diverse contexts including voting and matching markets.

Our paper addresses two issues that arise in ordinal mechanism design. To design social welfare maximizing mechanisms, one needs to be able to quantitatively measure the welfare of an outcome which is not clear in the ordinal setting. Second, since the impossibility results of Gibbard and Satterthwaite [14, 25] force one to move to randomized mechanisms, one needs a more nuanced notion of truthfulness.

We propose *rank approximation* as a metric for measuring the quality of an outcome, which allows us to evaluate mechanisms based on worst-case performance, and *lex-truthfulness* as a notion of truthfulness for randomized ordinal mechanisms. Lex-truthfulness is stronger than notions studied in the literature, and yet flexible enough to admit a rich class of mechanisms *circumventing classical impossibility results*. We demonstrate the usefulness of the above notions by devising lex-truthful mechanisms achieving good rank-approximation factors, both in the general ordinal setting, as well as structured settings such as *(one-sided) matching markets*, and its generalizations, *matroid* and *scheduling* markets.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems; G.2 [**Discrete Mathematics**]; J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*

Keywords

Algorithmic game theory; Mechanism design; Ordinal preferences; Computational social choice theory; Social welfare and rank approximation; Truthfulness for randomized mechanisms; Linear programming; Matroids

1. INTRODUCTION

A central problem in social choice theory and mechanism design is that of choosing a "good" outcome by aggregating individuals' private preferences over outcomes, where individuals are rational agents. A *mechanism* implementing a *social choice function* (SCF) needs to elicit the preferences of agents in a truthful fashion, that is, in a way such that no agent may benefit by misreporting his preferences.

In this paper, we study mechanism-design problems in *ordinal* settings, wherein the preferences are described by orderings over the set of outcomes. This is in contrast with the *cardinal* setting, wherein an agent specifies a *value* to each outcome (which determines his preferences). Ordinal settings reduce the "informational burden" on an agent in the sense that he only needs to be able to compare outcomes rather than assign values to outcomes justifying his preferences. It is not hard to imagine settings where the former comparison task is easier, and more aptly describes the situation: examples span the spectrum between electoral settings and the setting of allocating dormitory rooms to students.

Two immediate issues arise in ordinal mechanism design. A typical mechanism-design goal is to maximize *social welfare*, but in order to approach this goal in ordinal settings, one needs to first be able to *quantitatively* measure the social-welfare value of an outcome. Second, since the Gibbard-Satterthwaite (GS) impossibility result [14, 25] precludes non-trivial deterministic truthful mechanisms, one is forced to move to *randomized mechanisms* for which one needs a more nuanced notion of truthfulness.

1.1 Our contributions

We propose a new framework for welfare-maximization and truthfulness for randomized ordinal mechanisms, and devise various near-optimal mechanisms in this framework. Our contributions are threefold.

1) We introduce a metric called *rank approximation* for measuring the quality of an outcome, which in turn allows us

*A single-column version is available from the CS arXiv [9].

†Supported in part by NSERC grant 327620-09, an NSERC Discovery Accelerator Supplement Award, and an Ontario Early Researcher Award.

ITCS'14, January 12–14, 2014, Princeton, New Jersey, USA.

ACM 978-1-4503-2698-8/14/01 ...$15.00.
http://dx.doi.org/10.1145/2554797.2554810.

to evaluate mechanisms in terms of their worst-case performance. We show that rank approximation is a robust notion that is appealing, and can be motivated, from various perspectives.

2) We propose a truthfulness notion called *lex-truthfulness* for randomized ordinal mechanisms. This is stronger than a notion studied in the literature, and yet flexible enough that it admits a rich class of mechanisms *bypassing classical impossibility results.* We provide a characterization result for lex-truthfulness, which we leverage to obtain lex-truthful mechanisms for various ordinal settings. We believe that this characterization will find application beyond the specific applications that we consider in this paper.

3) We demonstrate the usefulness of the above two notions by devising lex-truthful mechanisms achieving good rank-approximation factors both in the general ordinal setting, as well as structured settings such as *(one-sided) matching markets*, and its generalizations, *matroid* and *scheduling* markets.

We now elaborate on our contributions. Let n and m denote the number of agents and number of outcomes respectively, and $\succ_j$ denote agent j's ordering over outcomes, which we assume is strict and complete (i.e., for any two outcomes o, o', either $o \succ_j o'$ or vice versa).

Rank approximation (Section 3.1). We say that an outcome o has *rank approximation* α for preference profile $\succ$, if for *every* position r, the number of agents having o as one of their top-r outcomes is at least $\frac{1}{\alpha} \cdot \mathrm{maxrank}_r(\succ)$, where $\mathrm{maxrank}_r(\succ)$ denotes $\max_{\hat{o}}$(number of agents having $\hat{o}$ as one of their top-r outcomes). An *α-rank-approximation mechanism* is one that always returns an α-rank-approximate outcome. While the requirement of simultaneously approximating $\mathrm{maxrank}_r(\succ)$ for all r seems too stringent, and even the *existence* of an α-rank-approximate outcome o, for non-trivial α seems doubtful, promisingly (as we elaborate later), we can achieve a 2-rank-approximation for matching and matroid markets, and a randomized $O(\log n)$-rank-approximation for general ordinal settings.

Rank approximation is a natural, purely ordinal notion with various desirable properties. Consider any cardinal-utility profile $\vec{U} = (U_1, \ldots, U_n)$, where each U_j is consistent with $\succ_j$, that is, $U_j(o) > U_j(o')$ iff $o \succ_j o'$. Call such a utility profile *homogeneous*, if for all $r = 1, \ldots, m$, all U_js assign the same value to their r-th ranked outcome. An α-rank-approximation outcome o for $\succ$ is such that for *any* consistent homogeneous utility profile $\vec{U}$, its social welfare, $\sum_{j=1}^{n} U_j(o)$, for $\vec{U}$ is at least a $\frac{1}{\alpha}$-fraction of the optimum social welfare for $\vec{U}$. Thus, an α-rank-approximation mechanism *simultaneously yields an α-approximation to the optimum social-welfare for all consistent homogeneous utility profiles* (Theorem 3.1).

Consistent homogeneous utilities are also known as *scoring rules* [29]. A scoring rule assigns a score to each position and returns the outcome with highest total score; a prominent example is the *Borda rule*, which gives a score of $m-k$ to the k-th position. An outcome is α-approximate with respect to a scoring rule, if its score is at least a $\frac{1}{\alpha}$-fraction of the score of any other outcome. Translated to this setting, we obtain that an α-rank-approximation mechanism *simultaneously achieves an α-approximation to all scoring rules.* In other words, given an α-rank-approximation mechanism $\mathcal{M}$, one need not be overly concerned about which scoring rule is most suited to the problem, since $\mathcal{M}$ guarantees an α-approximation to all scoring rules!

To place these simultaneous-approximation bounds in perspective, it is useful to consider an even stronger notion: say that a mechanism has "strong welfare factor" α, if for every consistent (even non-homogeneous) cardinal-utility profile $\vec{U}$, the mechanism returns an α-approximation to the optimum social welfare for $\vec{U}$. Not surprisingly, this notion is too strong: it is easy to show that no mechanism (deterministic or randomized) can have any non-trivial strong welfare factor, even for matching markets.

Lex-truthfulness (Section 3.2). The classic impossibility results of [14, 25] show that the space of deterministic truthful mechanisms in general ordinal settings is extremely limited, forcing the move to randomized mechanisms. When seeking to define a notion of truthfulness for ordinal randomized mechanisms, one immediately encounters the following issue: *how should one extend an agent's preferences over outcomes to preferences over distributions of outcomes?* The usual approach in the economics literature is to use the *stochastic dominance* relation. Since this does not induce a total order over distributions, one obtains two notions of truthfulness: (i) *strong truthfulness* [15], where the truth-telling distribution stochastically dominates any distribution obtained via a misreport; and (ii) *weak truthfulness* [21, 7], where the truth-telling distribution is not stochastically dominated by any distribution obtained via a misreport. Gibbard [15] generalized [14, 25] to show that the space of strongly-truthful mechanisms in general ordinal settings is also limited, leaving weak-truthfulness as the only viable notion of truthfulness for randomized mechanisms.

We propose a new notion of truthfulness sandwiched (strictly) between the above two notions. A distribution $\mathbf{p}$ lex-dominates a distribution $\mathbf{q}$ with respect to ordering $\succ$, if, when considering outcomes in decreasing order of their ranking in $\succ$, at the first outcome o where $\mathbf{p}$ and $\mathbf{q}$ differ, $\mathbf{p}$ assigns a higher probability to o than $\mathbf{q}$. Note that lex-dominance induces a *total order* on distributions. We say that a mechanism is *lex-truthful* (LT) if the truth-telling distribution lex-dominates any distribution obtained by a misreport.

We show that lex-truthfulness provides us with ample flexibility in mechanism design and allows us to *circumvent Gibbard's impossibility theorem.* Call a social choice function (SCF) f *fully lex-truthfully (LT) implementable* if for all $\varepsilon > 0$, there exists a lex-truthful mechanism that agrees with f with probability at least $(1-\varepsilon)$ on every preference profile. We isolate a property of an SCF, which we call *pseudomonotonicity*, that *completely characterizes* LT-implementability of the SCF (Theorem 3.3). Roughly speaking, an SCF is pseudomonotone if for any preference profile, if an agent j changes his ordering without altering his top k choices, then the new outcome cannot both be a better outcome for j and a top-$(k+1)$ outcome for j (see Definition 5).

This characterization turns out to be instrumental in making lex-truthfulness an amenable notion to work with, and opens up a host of SCFs to full LT-implementation. We show that various rank-approximation SCFs that we devise for matching, matroid, and scheduling markets—including the 2-rank-approximation mechanism for matching markets mentioned earlier—are pseudomonotone. For general ordinal settings, we identify a rich class of pseudomonotone SCFs which includes the *plurality scoring rule*. Thus, *all* of these SCFs are fully LT-implementable. We view the characterization of lex-truthfulness via pseudomonotonicity as one of our main contributions, which we believe will find further applications.

Matching, matroid, and scheduling markets (Sections 4 and 5). In addition to general ordinal mechanism-design settings, we also consider various structured settings, and obtain lex-truthful mechanisms with good rank-approximation factors.

Our most-compelling results are for *matching markets* (Section 4), which are one of the most well-studied ordinal settings (see, e.g., the surveys [28, 1]). Here, we have n agents and m items, and outcomes are matchings of agents to items. Each agent has a strict preference over items, which induces preferences over matchings based on the item the agent is assigned in a matching. Observe that agents are indifferent over matchings that give them the same item. The room allocation problem is an instance of this market.

We devise a simple deterministic 2-rank-approximation pseudomonotone algorithm MaxMatch (Theorem 4.1), which is therefore fully LT-implementable. In contrast, we show in Appendix B that various common algorithms proposed for matching markets, such as the top-trading-cycle algorithm, randomized serial dictatorship, probabilistic serial, *all have rank approximation at least* $\Omega(\sqrt{n})$. We prove a matching lower bound of 2 on the rank-approximation factor of deterministic SCFs (Theorem 4.2), and obtain super-constant lower bounds on the rank-approximation factor achievable by deterministic truthful mechanisms.

The 2-rank-approximation for matching markets extends to *matroid markets* (Theorem 4.6), which is the generalization where we have a matroid on the agent-set for every item, and the (possibly multiple) agents assigned to an item are required to form an independent set in that item's matroid. Besides the increased modeling power of matroids, this turns out to be a key component of our algorithms for scheduling markets.

In Section 5, we consider scheduling markets. Here the agents are jobs that need to be assigned to machines. Each job has a *private* ordering over the machines, and a public processing time on each machine, and there is makespan bound T that limits the amount of time available on each machine. An outcome is a partial assignment of some jobs to machines satisfying the makespan bound. This can be viewed as the matching problem with a *knapsack constraint*. For parallel machines, we obtain an LT-mechanism that always returns an $O(\log n)$-rank-approximation schedule with $O(T)$ makespan, and we show that this bound is tight (Theorems 5.2 and 5.3). We also obtain an $O(\log n)$-rank approximation for unrelated machines (Theorem 5.4), albeit not via an LT mechanism.

1.2 Other related work

The conundrum of social welfare in ordinal mechanisms, which probably has its origins in the Condorcet paradox [11] that states that it may so happen that a majority of agents prefer outcome a to b, outcome b to c, and outcome c to a, was cemented by Arrow's impossibility theorem [4]. Subsequent to Arrow's result, most works in social choice theory has focused on Pareto optimality as the sole notion of efficiency for ordinal mechanisms.

Recent work, mostly in the CS literature, has led to a more nuanced notion of efficiency. Procaccia and Rosenchein [23] studied the strong welfare factor notion (that they call distortion), and noticed that deterministic mechanisms have unbounded distortion. Boutilier et al. [8] proposed randomized mechanisms and showed that the strong welfare factor is at most $O(\sqrt{m}\log^* m)$, if the consistent cardinal-utility profile is normalized. In contrast, our rank approximation results imply $O(\log n)$-approximate outcomes, but under a stronger restriction on the consistent cardinal utilities. The notion of approximations to scoring rules was studied by Procaccia [22] where he described strongly truthful mechanisms which 2-approximate Borda, but $O(\sqrt{m})$-approximate the plurality rule. In contrast, our (non-truthful) mechanism $O(\log n)$-rank approximates *any* scoring rule, and plurality can be arbitrarily well approximated by a lex-truthful mechanism.

Another notion of social welfare in ordinal mechanisms, called ordinal welfare factor (OWF), was recently proposed by Bhalgat et al [6]. A mechanism has OWF $\beta \in [0, 1]$ if for any outcome o, at least βn agents prefer the outcome returned by the mechanism to o. This is in fact a *quantification* of the notion of *popular* outcomes; an outcome is popular if a majority prefer it to any other fixed outcome. Note that popular outcomes have OWF of at least 0.5. A popular outcome may not exist, but a popular distribution over outcomes always does. Popular outcomes were studied by economists in the matching setting [13], and as *strict maximal lotteries* in the general setting [12, 18]; subsequently, a large body of literature has been developed by computer scientists on popular matchings [2, 17, 16, 19]. The notions of rank approximation and OWF (and therefore the notion of popularity) are incomparable. That is, there are outcomes with "good" OWF and "bad" rank approximation, and vice-versa.

Subsequent to the Gibbard-Satterthwaite result, researchers focused on design of randomized mechanisms. As mentioned above, this led to differing notions of truthfulness. Strong truthfulness was proposed by Gibbard [15]. Postlewaite and Schmeidler [21] proposed weak truthfulness and proved that no weakly truthful mechanism on 4 or more outcomes, can be (ex ante) Pareto optimal if agents are allowed to have priors on their (own) preferences. Subsequently, Aziz et al [5] removed the prior condition, but prove impossibility of only certain *kinds* of mechanism. We remark that our lex-truthful mechanisms, which are also weakly truthful, do not contradict these results, since our mechanisms are not Pareto optimal. However, our mechanisms are ε-implementations of Pareto-optimal SCFs, so they satisfy Pareto optimality with probability at least $1 - \varepsilon$. Thus, we bypass the above impossibility results while sacrificing a modicum of Pareto-optimality.

Matching markets are one of the most widely studied examples of the ordinal setting. There is a vast amount of literature, and we point to excellent surveys [24, 28, 1]. In Section B, we describe three well known mechanisms in this setting. These are the random serial dictatorship, Gale's top trading cycle algorithm [27], and the probabilistic serial (PS) mechanism [7]. The first two mechanisms are at least strongly truthful; PS is weakly truthful, and we show that it is lex-truthful as well. However, we show that all these three mechanisms have rank approximation as bad as $\Omega(\sqrt{n})$. In contrast, we obtain a fully LT-implementable 2-rank-approximation mechanism using our pseudomonotone 2-rank-approximation algorithm MaxMatch.

2. PRELIMINARIES

In the general *ordinal mechanism design* setting, we have a set N of n agents, and a set O of m outcomes (or alternatives). We use the terms agent and player interchangeably. Each agent $j \in N$ has a *private* complete preference list or ordering $\succeq_j$ over outcomes, that is, $o \succeq_j o'$ or $o' \succeq_j o$ for every $o, o' \in O$. This is typically referred to as *ordinal* utilities/preferences, to distinguish them from *cardinal* utilities wherein the utility function assigns a value to each outcome. Let Σ_j denote the publicly-known set of allowed preference lists for agent j, and $\Sigma := \prod_{j=1}^{n} \Sigma_j$. A preference profile is a combination $\succeq = (\succeq_1, \ldots, \succeq_n)$ of agents' preference lists. For $k \in \mathbb{Z}_+$, we use $[k]$ to denote the set $\{1, \ldots, k\}$. A preference list is called *strict*, and denoted $\succ$, if there are no indifferences: exactly one of $o \succ o'$ and $o' \succ o$ holds for every two distinct outcomes $o, o' \in O$. Given a strict preference $\succ$, we will sometimes say $o \succeq o'$ to denote that $o \succ o'$ or $o = o'$. Given a preference list $\succ$, let $\text{alt}(\succ, r) \in O$ denote the r-th ranked outcome in $\succ$, and $\text{pos}(\succ, o) \in [m]$ denote the rank of outcome o in $\succ$. For a tuple $x = (x_1, \ldots, x_n)$, we use x_{-j} to denote $(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n)$. Similarly, let $\Sigma_{-j} := \prod_{k \neq j} \Sigma_k$.

In addition to the general setting mentioned above, we consider three specific mechanism-design settings in this paper: one-sided *matching markets*, which have been studied extensively in the literature (see, e.g., [28, 1]) and two generalizations of these, *matroid markets* and *scheduling markets*, that we introduce.

Matching markets (Section 4). We nave n agents and m items. Each agent j has a strict preference $\succ_j$ over the m items. The outcomes are matchings of agents to items. We say that an outcome M assigns an agent j the "null" item $\emptyset$ to denote that he is not assigned an item in M; we set $i \succ_j \emptyset$ for every item i. An agent is indifferent between matchings M and M' if they allot him the same item (counting $\emptyset$ as an item), and otherwise, prefers M to M' if he prefers the item allotted to him in M to the item allotted to him in M'.

Matroid markets (Section 4.1). We again have n agents who have a strict preference over m items. We also have a matroid $M_i = (N, \mathcal{I}_i)$ on the set N of agents, for each item $i \in [m]$. An outcome is an allocation that assigns at most one item to each agent j such that, for each item i, the set of agents allotted item i is an independent set of M_i. Note that multiple agents may be allocated the same item. An agent's ordering over outcomes is induced by his ordering of the items as in the setting of matching-markets. It is easy to see that a matching market is the special case where M_i encodes that at most one agent may be assigned to item i.

Scheduling markets (Section 5). The agents are n jobs that need to be scheduled on m machines, where the machines are in general *unrelated*. Each job j possesses a *private* strict complete preference order $\succ_j$ over the machines, and has a *publicly-known* processing time p_{ij} on machine i. Furthermore, there is a bound T on the maximum load allowed on any machine (i.e., makespan). An outcome is an (partial) assignment of some jobs to machines that respects the makespan bound. The ordering over outcomes is induced by the ordering over machines as in the above two cases. The parallel machines setting is the case where $p_{ij} = p_j$ for every machine i and job j.

Note that in the above three markets, agents' preferences over outcomes are *not* strict; however, for each agent j, the outcome-set may be partitioned into *indifference classes* such that j is indifferent between the outcomes in an indifference class, and has a strict ordering over the indifference classes. Our framework and results apply to such settings with cosmetic notational changes (see "Settings with indifferences" in Section 3), but we stick for the most part to the setting of strict preferences for notational ease.

A *social choice function* (SCF) is a function $f : \Sigma \mapsto O$. In settings with no monetary transfers, there is no formal distinction between an SCF and a *deterministic* algorithm or *direct-revelation* mechanism, which maps the preference profile given by the agents' reported preference lists to an outcome. An SCF f is said to be *implementable* or *truthful* if for every player j, every $\succ_j, \succ'_j \in \Sigma_j$, and every $\succ_{-j} \in \Sigma_{-j}$, we have $f(\succ_j, \succ_{-j}) \succeq_j f(\succ'_j, \succ_{-j})$; that is, no agent benefits by misreporting his preference list.

A *randomized mechanism* is allowed to output a distribution (also called a *lottery*) over outcomes. Let $\mathcal{L}(O)$ denote the collection of distributions over the outcome-set O. A randomized mechanism is formally then a function mapping preference profiles to distributions in $\mathcal{L}(O)$. We sometimes refer to a mechanism that works with ordinal preferences as an ordinal mechanism.

Definition 1. A randomized mechanism $\mathcal{M}$ is said to ε-*implement* an SCF f (or that f is ε-implementable by $\mathcal{M}$), if $\mathbf{Pr}[\mathcal{M}(\succ) = f(\succ)] \geq 1 - \varepsilon$ for all $\succ \in \Sigma$, where the probability is over the random choices of $\mathcal{M}$. We say that a family $\{\mathcal{M}^{\varepsilon}\}$ of mechanisms *fully implements* f if for all $\varepsilon > 0$, $\mathcal{M}^{\varepsilon}$ ε-implements f. (This is in the same spirit as the notion of virtual implementation in Nash equilibrium [20, 3].)

Truthfulness for randomized mechanisms may be defined in various ways. The strongest notion is *universal truthfulness*, wherein a randomized truthful mechanism is a randomization (or mixture) over deterministic truthful mechanisms, where the mixture weights are input-independent. A somewhat weaker notion is obtained by considering the stochastic dominance relation. Given an ordering $\succ$ over O, and two lotteries $\mathbf{p}, \mathbf{q} \in \mathcal{L}(O)$, we say that $\mathbf{p}$ (first-order) *stochastically dominates* $\mathbf{q}$ with respect to $\succ$, if $\sum_{\ell \leq i} \mathbf{p}(\text{alt}(\succ, \ell)) \geq \sum_{\ell \leq i} \mathbf{q}(\text{alt}(\succ, \ell))$ for all $i = 1, \ldots, m$. Since stochastic dominance does not induce a total ordering on $\mathcal{L}(O)$, this yields two notions of truthfulness that have been studied in the literature.

Definition 2. A randomized mechanism $\mathcal{M}$ is said to be:

- *strongly truthful* [15]: if $\mathcal{M}(\succ_j, \succ_{-j})$ *stochastically dominates* $\mathcal{M}(\succ'_j, \succ_{-j})$ with respect to $\succ_j$, for all j, all $\succ_j, \succ'_j \in \Sigma_j$, and all $\succ_{-j} \in \Sigma_{-j}$.
- *weakly truthful* [21, 7]: if $\mathcal{M}(\succ_j, \succ_{-j})$ *is not stochastically dominated* by $\mathcal{M}(\succ'_j, \succ_{-j})$ with respect to $\succ_j$, for all j, all $\succ_j, \succ'_j \in \Sigma_j$, and all $\succ_{-j} \in \Sigma_{-j}$.

A universally truthful mechanism is also strongly truthful, and in fact, this inclusion is strict (Theorem 3.2). Gibbard [15] extended the impossibility result of [14, 25] to show that the space of strongly truthful mechanisms is also rather limited. A deterministic mechanism is: (i) *dictatorial* if there exists $j \in N$ such that the mechanism's output is always j's top choice; and (ii) *duple* if the mechanism's range $f(\Sigma)$ consists of at most two outcomes. A (deterministic or randomized) mechanism is *unilateral* if there exists some fixed $j \in N$ such that the mechanism's output depends only on j's (reported) preference list.

THEOREM 2.1. *(Gibbard-Satterthwaite and Gibbard impossibility results) (i) If $m \geq 3$ and $f(\Sigma) = O$, then f is truthful iff it is dictatorial. (ii) Any strongly truthful mechanism is a mixture of truthful unilateral and deterministic truthful duple mechanisms with input-independent mixture weights.*

Theorem 2.1 leaves weak truthfulness as the only notion that potentially allows for some sophisticated mechanisms. In Section 3.2, we propose a stronger notion of truthfulness and show that this is flexible enough that one can *bypass Gibbard's impossibility result* and obtain various interesting mechanisms including, in particular, mechanisms that yield "good" social welfare under the metric we introduce in Section 3.1.

3. RANK APPROXIMATION AND LEX-TRUTHFULNESS

3.1 Welfare in ordinal settings: rank approximation

We introduce a notion of social welfare that we call *rank approximation*. Given a preference profile $\succ = (\succ_1, \ldots, \succ_n)$, the i-rank of an outcome $o \in O$ in $\succ$, denoted $\mathrm{rank}_i(o; \succ)$, is the number of agents having o in their top i choices: $\mathrm{rank}_i(o;\succ) = \big|\{j : \mathrm{pos}(\succ_j, o) \leq i\}\big|$. Define $\mathrm{maxrank}_i(\succ) := \max_{o \in O} \mathrm{rank}_i(o; \succ)$.

Definition 3. A randomized mechanism $\mathcal{M}$ is an *α-rank-approximation* mechanism, if for every preference profile $\succ$, we have $\mathbf{E}\big[\mathrm{rank}_i(\mathcal{M}(\succ);\succ)\big] \geq \mathrm{maxrank}_i(\succ)/\alpha$ for *all* $i = 1, \ldots, m$, where the expectation is taken over the random choices of $\mathcal{M}$. We say that α is the rank-approximation factor of $\mathcal{M}$.

As mentioned in the Introduction, rank approximation is an appealingly robust notion from various perspectives. A utility function U is consistent with a preference ordering $\succ$ if $U(o) > U(o')$ whenever $o \succ o'$. A collection of cardinal utility functions $(U_1, \ldots, U_n)$ consistent with a preference profile $\succ$ is called *homogeneous* if for all $i \in [m]$, the value that an agent assigns to his i-th choice is the same across all agents, that is, $U_j(\mathrm{alt}(\succ_j, i)) = U_{j'}(\mathrm{alt}(\succ_{j'}, i))$ for all $i \in [m], j, j' \in N$.

An α-rank-approximation mechanism yields an α approximation to social welfare for *any* homogeneous cardinal-utility profile consistent with the agents' preferences.

THEOREM 3.1. *Let $\mathcal{M}$ be an α-rank-approximation randomized mechanism. Then, for every preference profile $\succ$, we have $\mathbf{E}\big[\sum_{j\in N} U_j\big(\mathcal{M}(\succ)\big)\big] \geq \frac{1}{\alpha} \cdot \max_{o\in O} \sum_{j \in N} U_j(o)$ for any homogeneous utility profile $(U_1, \ldots, U_n)$ consistent with $\succ$.*

PROOF. Let $\mathbf{p} = \mathcal{M}(\succ)$. Let $U(i)$ be the common value of $U_j(\mathrm{alt}(\succ_j, i))$. Define $\mathrm{rank}_0(o;\succ) = 0$ for all $o \in O$, and $U(m+1) = 0$. Let $o^* = \mathrm{argmax}_{o \in O} \sum_{j\in N} U_j(o)$. Then $\mathbf{E}\big[\sum_{j \in N} U_j\big(\mathcal{M}(\succ)\big)\big]$ is

$$\begin{aligned}
&\sum_{o \in O} \mathbf{p}(o) \sum_{i=1}^{m} \big(\mathrm{rank}_i(o;\succ) - \mathrm{rank}_{i-1}(o;\succ)\big) U(i) \\
&= \sum_{i=1}^{m} \Big(\sum_{o\in O} \mathbf{p}(o)\, \mathrm{rank}_i(o;\succ)\Big)\big(U(i) - U(i+1)\big) \\
&\geq \frac{1}{\alpha} \cdot \sum_{i=1}^{m} \mathrm{rank}_i(o^*;\succ)\big(U(i) - U(i+1)\big) \\
&= \frac{1}{\alpha} \cdot \sum_{j \in N} U_j(o^*).
\end{aligned}$$

□

Consistent homogeneous utilities may be equivalently viewed as a *scoring rule*; Viewed from this perspective, Theorem 3.1 shows that an α-rank-approximation mechanism *simultaneously achieves an α-approximation to all scoring rules.*

In fact, rank approximation satisfies an even more general robustness property. Associate with each outcome o an m-vector called its *histogram*, given by $\mathrm{hist}(o;\succ) = \{\mathrm{rank}_i(o;\succ)\}_{i \in [m]}$. Define the g-value of an outcome o to be $g\big(\mathrm{hist}(o;\succ);\succ\big)$, where $g(x;\succ) := \min_{i\in[m]} \frac{x_i}{\mathrm{maxrank}_i(\succ)}$. It is not hard to see that g is a concave function of x and non-decreasing in each coordinate. A deterministic α-rank-approximation mechanism outputs an outcome with g-value at least $\frac{1}{\alpha}$.

Now suppose $h(x;\succ)$ is *any* concave non-decreasing function and we measure the value of an outcome o by $h\big(\mathrm{hist}(o;\succ);\succ\big)$. This yields a natural SCF f^h, where $f^h(\succ)$ is $\mathrm{argmax}_{o' \in O} h(o';\succ)$. Note that scoring rules correspond to the special case where $h(\cdot)$ is linear with all coefficients non-negative. Analogous to α-rank-approximation, we can define an SCF f' to be an α-approximation for f^h if (h-value of $f'(\succ)) \geq \frac{1}{\alpha} \cdot$ (h-value of $f^h(\succ)$) for all $\succ$.

An α-rank-approximation mechanism *simultaneously* achieves an α-approximation mechanism for *all* such histogram-based concave SCFs: if o is the outcome returned, we get $g\big(\mathrm{hist}(o;\succ);\succ\big) \geq \frac{1}{\alpha}$, and therefore $\mathrm{hist}(o;\succ) \geq \frac{1}{\alpha} \cdot \mathrm{hist}(o';\succ)$ coordinate-wise for any $o' \in O$. Since h is non-decreasing and concave, this implies that $h\big(\mathrm{hist}(o;\succ);\succ\big) \geq \frac{1}{\alpha} \cdot \max_{o' \in O} h\big(\mathrm{hist}(o';\succ);\succ\big) = \frac{1}{\alpha} \cdot f^h(\succ)$.

3.2 Truthfulness for randomized ordinal mechanisms: lex-truthfulness

We propose a new notion for truthfulness relying on lexicographic ordering. Given an ordering $\succ$ over O, and two lotteries $\mathbf{p} \neq \mathbf{q} \in \mathcal{L}(O)$, $\mathbf{p}$ *lexicographically dominates* $\mathbf{q}$ with respect to $\succ$, if there exists $i \in [m]$ such that $\mathbf{p}(\mathrm{alt}(\succ, i)) > \mathbf{q}(\mathrm{alt}(\succ, i))$ and $\mathbf{p}(\mathrm{alt}(\succ, \ell)) = \mathbf{p}(\mathrm{alt}(\succ, \ell))$ for all $\ell = 1, \ldots, i-1$. Note that lex-dominance imposes a *total order* on $\mathcal{L}(O)$. This motivates the following definition of truthfulness.

Definition 4. A randomized mechanism $\mathcal{M}$ is called *lex-truthful* (LT) if for all $j \in N$, all $\succ_j \succ'_j \in \Sigma_j$, and all $\succ_{-j}$, we have that either $\mathcal{M}(\succ_j, \succ_{-j}) \equiv \mathcal{M}(\succ'_j, \succ_{-j})$, or $\mathcal{M}(\succ_j, \succ_{-j})$ *lexicographically dominates* $\mathcal{M}(\succ'_j, \succ_{-j})$ with respect to $\succ_j$.

Observe that if $\mathbf{p}$ stochastically dominates $\mathbf{q}$, then $\mathbf{p}$ lex-dominates $\mathbf{q}$ as well. Since lex-dominance is a total order, this implies that if $\mathbf{p}$ lex-dominates $\mathbf{q}$, then $\mathbf{q}$ cannot stochastically dominate $\mathbf{p}$. We obtain the following hierarchy between the various notions of truthfulness for randomized ordinal mechanisms.

THEOREM 3.2. *Let* UnivT, StrongT, WeakT, LexT *denote the classes of universally-, strongly-, weakly-, and lex-truthful mechanisms respectively. Then* $\mathsf{UnivT} \subsetneq \mathsf{StrongT} \subsetneq \mathsf{LexT} \subsetneq \mathsf{WeakT}$.

We defer the proof of Theorem 3.2 to Appendix A. We shorten "implementable by a lex-truthful mechanism" to "lex-truthfully (LT) implementable" in the sequel. We show that lex-truthful implementability is equivalent to a property of the social-choice function that we call *pseudomonotonicity*. This characterization immediately opens up a host of SCFs that are fully LT-implementable. We heavily exploit this in Sections 4 and 5 to show that the rank-approximation SCFs that we devise for various problems are fully LT-implementable. In Section 6, we leverage this to show that an interesting class of SCFs in general ordinal settings are fully LT-implementable.

Definition 5. A social choice function f is *pseudomonotone* (or satisfies *pseudomonotonicity*) if the following holds. Consider any player j, $\succ_{-j}\in \Sigma_{-j}$, and $\succ_j, \succ'_j \in \Sigma_j$. Let $o = f(\succ)$ and $o' = f(\succ')$. Then, either (i) $o \succeq_j o'$, or (ii) there is an outcome o'' such that $o'' \succ_j o'$ and $\text{pos}(\succ_j, o'') < \text{pos}(\succ'_j, o'')$.

A useful way to view pseudomonotonicity is as follows: if a player's deviation leaves his first k preferences unaltered, then the deviation cannot both yield him a better outcome *and* a top-$(k+1)$ outcome.

THEOREM 3.3. *(i) Let f be a pseudomonotone SCF. Then f is ε-implementable by a lex-truthful mechanism for any $\varepsilon > 0$; that is, f is fully lex-truthfully implementable.*
(ii) Conversely, if f is ε-LT implementable for some $\varepsilon < \frac{1}{2}$, then f is pseudomonotone.

PROOF. First consider part (i). Given $\varepsilon > 0$, one can find $\varepsilon_1 > \varepsilon_2 > \cdots > \varepsilon_m > 0$ such that $\sum_i \varepsilon_i = \varepsilon$. Consider the randomized mechanism $\mathcal{M}$ that on input $\succ$, returns $f(\succ)$ with probability $(1-\varepsilon)$, and with probability ε it chooses a random agent a and returns his i-th preference with probability $\varepsilon_i/\varepsilon$.

It is clear by definition that $\mathcal{M}$ ε-implements f. To prove lex-truthfulness, fix an agent j and consider any $\succ' = (\succ'_j, \succ_{-j})$, where $\succ'_j \neq \succ_j$. Let $o = f(\succ)$ and let $o' = f(\succ')$. Also let $\mathbf{p} = \mathcal{M}(\succ)$, $\mathbf{q} = \mathcal{M}(\succ')$. Let $\mathbf{1}(A)$ be 1 if A is true, and 0 otherwise. For any outcome $\hat{o}$, we have

$$\begin{aligned}\mathbf{p}(\hat{o}) - \mathbf{q}(\hat{o}) &= \frac{1}{n}\left(\varepsilon_{\text{pos}(\succ_j,\hat{o})} - \varepsilon_{\text{pos}(\succ'_j,\hat{o})}\right) \\ &\quad + \mathbf{1}(\hat{o} = o)\cdot(1-\varepsilon) - \mathbf{1}(\hat{o} = o')\cdot(1-\varepsilon).\end{aligned}$$

Considering outcomes in the preference order of $\succ_j$, let o'' be the first outcome such that $\text{pos}(\succ_j, o'') \neq \text{pos}(\succ'_j, o'')$. Then $\text{pos}(\succ_j, o'') < \text{pos}(\succ'_j, o'')$. By pseudomonotonicity of f, we know that $o \succeq_j o'$ or $o'' \succ_j o'$. In the latter case, we have $\mathbf{p}(\hat{o}) - \mathbf{q}(\hat{o}) \geq 0$ for all $\hat{o} \succeq_j o''$ and $\mathbf{p}(o'') - \mathbf{q}(o'') > 0$, so we are done. In the former case, if $o = o'$ or $o'' \succeq_j o$, then the same argument holds. So suppose $o \succ_j o''$ and $o \succ_j o'$. Then $\mathbf{p}(\hat{o}) - \mathbf{q}(\hat{o}) \geq 0$ for all $\hat{o} \succeq_j o$ and $\mathbf{p}(o) - \mathbf{q}(o) > 0$, so again we are done.

We now prove part (ii). Let $\mathcal{M}$ be an LT mechanism that ε-LT implements f, where $\varepsilon < \frac{1}{2}$. Suppose for a contradiction, there is some agent j, and $\succ = (\succ_j, \succ_{-j})$ and $\succ' = (\succ'_j, \succ_{-j})$ such that $o = f(\succ)$ and $o' = f(\succ')$ violate the conditions for pseudomonotonicity. That is, we have $o' \succ_j o$ and for every outcome $o'' \succ_j o'$, we have $\text{pos}(\succ_j, o'') \geq \text{pos}(\succ'_j, o'')$. This means that $\text{pos}(\succ_j, o'') = \text{pos}(\succ'_j, o'')$ for all $o'' \succ_j o'$. Let $\mathbf{p} = \mathcal{M}(\succ)$ and $\mathbf{q} = \mathcal{M}(\succ')$.

Since $\mathcal{M}$ ε-LT implements f, we have $\mathbf{p}(o') \leq \varepsilon$ and $\mathbf{q}(o') \geq 1-\varepsilon$, so $\mathbf{p}(o') < \mathbf{q}(o')$. Let $o_1, \ldots, o_r$ be the outcomes o'' such that $o'' \succ_j o'$ listed in decreasing preference order according to $\succ_j$. Since $\text{pos}(\succ_j, o'') = \text{pos}(\succ'_j, o'')$ for all $o'' \succ_j o'$, we have $o_\ell \succ'_j o'$ for all $\ell \in [r]$, and the ordering of the o_ℓs is the same in $\succ_j$ and $\succ'_j$. We claim that $\mathbf{p}(o_\ell) = \mathbf{q}(o_\ell)$ for all $\ell = 1, \ldots, r$, which contradicts the fact that $\mathcal{M}$ is lex-truthful.

We prove the claim by induction on ℓ. Considering $\succ_j$ to be j's true preference list, we must have $\mathbf{p}(o_1) \geq \mathbf{q}(o_1)$, and considering $\succ'_j$ to be j's true preference list, we must have $\mathbf{q}(o_1) \geq \mathbf{p}(o_1)$. Suppose that $\mathbf{p}(o_k) = \mathbf{q}(o_k)$ for $k = 1, \ldots, \ell-1$. Again considering $\succ_j$ and $\succ'_j$ be j's true preference lists in turn, we obtain that $\mathbf{p}(o_\ell) = \mathbf{q}(o_\ell)$. $\square$

Settings with indifferences.

As noted earlier, many of the settings we consider involve non-strict preferences. In these settings, the outcome set is partitioned into *indifference classes* $O_1^j, \ldots, O_{m_j}^j$ for each agent j. Agent j is indifferent between any two outcomes in the same indifference class, and has a strict complete ordering over his indifference classes that specifies his ordering between two outcomes in different classes. Formally, given $\succeq_j \in \Sigma_j$, we define $\text{pos}(\succeq_j, o) = r \in [m_j]$ if o lies in the indifference class of j ranked r under $\succeq_j$, and $\text{alt}(\succeq_j, r) \subseteq O$ is now the indifference class of j ranked r under $\succeq_j$ (that is, $\{o : \text{pos}(\succeq, o) = r\}$). The preferences induced over outcomes are then: $o \succeq_j o'$ iff $\text{pos}(\succeq_j, o) \leq \text{pos}(\succeq_j, o')$, and $o \succ_j o'$ iff $\text{pos}(\succeq_j, o) < \text{pos}(\succeq_j, o')$. Say that $o \sim_j o'$ if o and o' belong to the same indifference class of j.

One requires mostly notational changes to extend our framework and results to this more-general setting. With the above notation in place, the definitions of $\text{rank}_i(o; \succeq)$ (as $|\{j : \text{pos}(\succeq_j, o) \leq i\}|$, $\text{maxrank}_i(\succ)$, rank-approximation (Definition 3) and pseudomonotonicity (Definition 5) remain unchanged.

We extend lex-dominance and lex-truthfulness as follows. Since players have indifference classes it is not meaningful to consider probabilities assigned to individual outcomes; instead, we consider the total probability assigned to an indifference class. Given a lottery $\mathbf{p} \in \mathcal{L}(O)$ and $S \subseteq O$, define $\mathbf{p}(S) := \sum_{o \in S} \mathbf{p}(o)$. Given $\succeq_j \in \Sigma_j$, and lotteries $\mathbf{p} \neq \mathbf{q} \in \mathcal{L}(O)$, we say that $\mathbf{p}$ lex-dominates $\mathbf{q}$ with respect to $\succeq_j$ if there exists $r \in [m_j]$ such that $\mathbf{p}(\text{alt}(\succeq_j, r)) > \mathbf{q}(\text{alt}(\succeq_j, r))$ and $\mathbf{p}(\text{alt}(\succeq_j, \ell)) = \mathbf{q}(\text{alt}(\succeq_j, \ell))$ for all $\ell = 1, \ldots, r-1$. With this definition of lex-dominance, lex-truthfulness remains as defined in Definition 4.

We can now mimic the proof of Theorem 3.3 to prove the following analogue for the above setting, showing that pseudomonotonicity is necessary and sufficient for full LT-implementability. The proof appears in Appendix A.

THEOREM 3.4. *(i) Let f be a pseudomonotone SCF. Then f is fully lex-truthfully implementable.*

(ii) Conversely, if f is ε-lex-truthfully implementable for any $\varepsilon < \frac{1}{2}$, then f is pseudomonotone.

4. MATCHING MARKETS

Recall that in a matching market there are n agents and m items, and outcomes are matchings of agents to items. Each agent j has a strict total ordering over items, which induces his preferences over outcomes: j prefers outcome o to o' if he prefers his allotted item in o to the one in o'.

We show in Section B that various common mechanisms all have bad rank approximation. In contrast, we devise a simple deterministic algorithm, MaxMatch, that is a pseudomonotone, 2-rank-approximation algorithm (Theorem 4.1). We complement this by showing two lower bounds. Theorem 4.2 shows that 2 is the best rank approximation achievable by *any* deterministic algorithm, proving the tightness of our positive result. Next, Theorems 4.3 and 4.4 demonstrate limitations of *deterministic* truthful mechanisms for matching markets by showing that such mechanisms cannot achieve any constant rank approximation.

Algorithm MaxMatch .

Fix a tie-breaking rule over agents. On input $\succ$, MaxMatch allocates items to agents in m stages. In stage r, we consider the bipartite graph G_r with agents and items as vertices, and an edge from agent j to item i, if i is a top-r item of agent j. Note that $\text{maxrank}_r(\succ)$ is precisely the size of the maximum matching in G_r. Let M denote the current matching of agents to items (which is $\emptyset$ when $r = 1$), which is a matching in G_r. We maintain that at he beginning of stage r, M is a *maximal matching* in G_{r-1}; observe this is true when $r = 1$. Since M is a *maximal matching* in G_{r-1}, then any agent has an edge to at most one item in $G_r \setminus M$. For every unmatched item i that has non-zero degree in $G_r \setminus M$ (that is, i is a top-r item of some agent that is currently unmatched) we use our tie-breaking rule to pick an agent $j \in G_r \setminus M$; we assign item i to j and update M. Thus, M is updated to a maximal matching in G_r. We output the matching at the end of m stages.

THEOREM 4.1. MaxMatch *a pseudomonotone, 2-rank approximation algorithm for matching markets, and hence is fully LT-implementable.*

PROOF. The 2-rank-approximation guarantee of MaxMatch follows immediately from the fact that MaxMatch maintains a maximal matching in the "top-r" graph G_r for all r, and the size of any maximal matching is at least half the size of a maximum matching, and thus at least $\text{maxrank}_r(\succ)/2$.

Fix an agent j. Suppose that j deviates from $\succ_j$ to $\succ'_j$ without altering his top-r items and their ordering, that is, $\text{alt}(\succ_j, \ell) = \text{alt}(\succ'_j, \ell)$ for all $\ell = 1, \dots, r$, and $\text{pos}(\succ'_j, i) > r + 1$ for $i = \text{alt}(\succ_j, r + 1)$. Let $\succ = (\succ_j, \succ_{-j})$ and $\succ' = (\succ'_j, \succ_{-j})$. Since the other agents' inputs have not changed, MaxMatch$(\succ)$ and MaxMatch$(\succ')$ proceed identically up to the end of stage r. So if j has been assigned an item by this time (which happens in both runs) we are done. Otherwise, in MaxMatch$(\succ')$, all of j's top-r items are unavailable, and since j demotes i in $\succ'$, edge (j, i) does not belong to the graph G_{r+1} constructed in stage $r + 1$; so j does not obtain i or a top-r item under input $\succ'$. This proves pseudomonotonicity. $\square$

THEOREM 4.2. *No deterministic algorithm for matching markets can have rank-approximation $2 - \epsilon$, for any $\epsilon > 0$.*

PROOF. We create an instance with $n = 2K - 1$ players and items, where $\frac{K}{2K-1} < \frac{1}{2-\epsilon}$. We specify the first K preferences of the players; the remaining preferences may be set arbitrarily. Let $\succ$ denote the resulting input (with arbitrary remaining preferences). Since this is the only input we consider, we drop the $\succ$ in $\text{rank}_r(o; \succ)$ and $\text{maxrank}_r(\succ)$ in the sequel.

- For $r = 2, \dots, K - 1$, the r-th preference of a player j is item: $\begin{cases} r & ; \quad \text{if } j = r, \\ K + r - 1 & ; \quad \text{if } j = r - 1, \\ r - 1 & ; \quad \text{otherwise.} \end{cases}$
- The first preference of a player j is: item 1 if $j = 1$, and item n otherwise.
- The K-th preference of a player j is: item K if $j = K-1$, and item $K - 1$ otherwise.

First, we claim that $\text{maxrank}_r \geq 2r$ for all $r \in [K-1]$. For $r = 1$, this is achieved by matching player 1 to item 1, and an arbitrary other player to item n. For $r = 2, \dots, K - 1$, this is achieved by matching player r to item r, each player $j \in [r - 1]$ to item $K + j$, matching one player from $\{r + 1, \dots, n\}$ to item n and $r - 1$ other arbitrary players from $\{r + 1, \dots, n\}$ arbitrarily to items in $[r - 1]$. Note that each player is matched to a top-r item in this matching. Also, $\text{maxrank}_K = n$. This is achieved by matching player $K-1$ to item K, each player $j \in [K - 2]$ to item $K + j$, matching one player from $\{K, \dots, n\}$ to item n, and the remaining $K - 1$ players from $\{K, \dots, n\}$ arbitrarily to items in $[K - 1]$.

Now fix a matching o. We show that if $\text{rank}_r(o) > \text{maxrank}_r/2$ for $r = 1, \dots, K - 1$, then we must have $\text{rank}_K(o) \leq K < \text{maxrank}_K/(2-\epsilon)$. Thus, we cannot have $\text{rank}_r(o) \geq \text{maxrank}_r/(2-\epsilon)$ for all $r \in [K]$.

We show by induction on r that if $\text{rank}_\ell(o) > \text{maxrank}_\ell/2$ for all $\ell \in [r]$, where $r < K$, then o must match player ℓ to item ℓ for all $\ell \in [r]$. For the base case, if $\text{rank}_1(o) > \text{maxrank}_1/2 \geq 1$, then o must match player 1 to item 1 since all other players have item n as their top item. For the induction step, suppose that $\text{rank}_\ell(o) > \text{maxrank}_\ell/2$ for all $\ell \in [r]$, where $1 < r < K$. Then, by the induction hypothesis, we know that o matches player ℓ to item ℓ for all $\ell \in [r-1]$. We require that $\text{rank}_r(o) \geq r+1$. Examining the preferences of the players in $\{r, \dots, n\}$, we see that for player r, items r and n are the only unmatched items in his top-r list, and for a player $j \in \{r + 1, \dots, n\}$, item n is the only unmatched item in j's top-r list. Therefore, $\text{rank}_r(o) \geq r+1$ is only possible if o matches player r to item r.

Given the above claim, for players $j = K, \dots, n$, item n is the only unmatched item in their top-K list, so $\text{rank}_K(o) \leq K$. $\square$

We now show that randomization is necessary to achieve good rank approximation via truthful mechanisms. As a warm up, we first prove a lower bound of $n - 1$ on the rank-approximation factor achievable by truthful *no-bossy* mechanisms [25]. A *no-bossy mechanism* for matching markets is one where no agent can change his preference and modify the outcome without also modifying his own allocation.

Suppose there are n items. Let $\succ^* := (1, 2, \dots, n)$ denote the ordering where item i is the i-th ranked item, for all $i \in [n]$. Let $\succ^* \circ (k - 1, 1)$, denote the preference list that is identical to $\succ^*$ except that items $(k - 1)$ and 1 are swapped. That is, $(k - 1)$ is the top-item, 1 is the kth-choice, and item i is the i-th choice for all $i \neq 1, k - 1$. Given n agents and any set $S \subseteq \{2, 3, \dots, n\}$, let $\succ^S$ be the preference profile where each agent $k \in S$ has preference $\succ^* \circ(k-1,1)$, while each $k \notin S$ has preference $\succ^*$. Thus, $\succ^\emptyset$ is the preference profile where every agent has the same preference order $\succ^*$ over items. For notational convenience, we think of a player who is not assigned an item as being assigned item $n+1$, which is lower ranked than any (true) item in $[n]$.

THEOREM 4.3. *No deterministic truthful no-bossy mechanism for matching markets can have rank-approximation smaller than* $(n-1)$.

PROOF. We consider a matching market with n (agents and) items. Let $\mathcal{M}$ be any deterministic truthful no-bossy mechanism. Suppose that $\mathcal{M}(\succ^\emptyset)$ assigns items to N players. By renaming players if necessary, we may assume that $\mathcal{M}(\succ^\emptyset)$ assigns item i to player i for all $i \in [N]$, and item $n+1$ to the remaining players.

Consider the input $\succ^{\{k\}}$. We claim that $\mathcal{M}(\succ^{\{k\}}) = \mathcal{M}(\succ^\emptyset)$. *Due to no-bossiness*, it suffices to show that agent k's allocation is the same in $\mathcal{M}(\succ^{\{k\}})$ and $\mathcal{M}(\succ^\emptyset)$. Suppose agent k obtains item i in $\mathcal{M}(\succ^{\{k\}})$. Invoking truthfulness when k's true preference list is $\succ^*$ (and the other players' preference lists are $\succ^*$), we obtain that $k \succeq^* i$, that is, $k \leq i$. Similarly, if k's true preference list were $\succ^* \circ(k-1,1)$, then truthfulness dictates that $i \leq k$. Hence, we have $i = k$.

The above argument can be generalized to show that for any $S \subseteq [n]$, we have $\mathcal{M}(\succ^S) = \mathcal{M}(\succ^{S\setminus k}) = \mathcal{M}(\succ^\emptyset)$ for all $k \in S$. In particular, $\mathcal{M}(\succ^{\{2,\ldots,n\}})$ assigns item i to player i for all $i \in [n]$. Now, under the preference profile $\succ^{\{2,\ldots,n\}}$, at most one agent, agent 1, gets his top choice; however, assigning every player $j > 1$ item $j-1$ yields an outcome where $n-1$ agents get their top choice. $\square$

While no-bossiness was crucial above, we show via a more sophisticated argument that no deterministic truthful mechanism can obtain constant rank approximation.

THEOREM 4.4. *Every deterministic truthful mechanism has rank approximation* $\Omega\left(\frac{\log\log n}{\log\log\log n}\right)$.

PROOF. Let n be large enough so that $K := \left\lfloor \frac{\log\log n}{\log\log\log n} \right\rfloor - 2 \geq 1$. We show that on instances with n (agents and) items, no deterministic truthful mechanism can have rank approximation better than K.

As before, if $\mathcal{M}(\succ^\emptyset)$ assigns items to N players, we may assume that it matches agent i to item i for $i \in [N]$, and the remaining players are unassigned (i.e., assigned item $n+1$). Given agents $\{a_1, \ldots, a_k\}$ and integers $r_1, \ldots, r_k \geq 1$, we let $\succ^{(a_1,r_1),(a_2,r_2),\ldots,(a_k,r_k)}$ denote the preference profile where all agents other than these a_ℓ's have preference order $\succ^*$, while each a_ℓ has preference order $\succ^* \circ(r_\ell, 1)$). That is, a_ℓ's top choice is item r_ℓ, his r_ℓ-th choice is item 1, and his i-th choice is item i for all $i \neq 1, r_\ell$. We show that there exist agents $a_1, \ldots, a_K$ and *distinct* integers $r_1, \ldots, r_K \in [K]$, such that, in $\mathcal{M}(\succ^{(a_1,r_1),\ldots,(a_K,r_K)})$, every agent $a_1, \ldots, a_K$ gets an item whose index is larger than K. Since all other agents have the same top item, the number of agents getting their top item is at most 1. This proves that the rank approximation is at least K, since assigning item r_ℓ to agent a_ℓ for all $\ell \in [K]$, yields an outcome where K agents obtain their top-choice item.

To find these K agents, we proceed in K stages. In stage ℓ, we will have a subset S_ℓ of agents with $|S_\ell| \geq \ell$ having the following property. For any $t < \ell$, any t agents $\{a_1, \ldots, a_t\} \subseteq S_\ell$, and for any t distinct integers $r_1, \ldots, r_t \in [K]$, $\mathcal{M}(\succ^{(a_1,r_1),\ldots,(a_t,r_t)})$ allocates all agents in S_ℓ an item indexed larger than K.

Note that if we reach stage K, then we are done due to the following reason. Consider any K agents $a_1, \ldots, a_K \in S_K$ and any K distinct integers $r_1, \ldots, r_K \in [K]$. Consider any index $\ell \in [K]$. Let $\succ' = \succ^{(a_1,r_1),\ldots,(a_{\ell-1},r_{\ell-1}),(a_{\ell+1},r_{\ell+1}),(a_{K-1},r_{K-1})}$ and $\succ = \succ^{(a_1,r_1),\ldots,(a_K,r_K)}$. We know that $\mathcal{M}(\succ')$ allocates all agents in S_K an item indexed larger than K. This also implies that $o := \mathcal{M}(\succ)$ allocates a_ℓ an item indexed larger than K, otherwise given the preference profile $\succ'$, player ℓ has an incentive to deviate from his preference list $\succ^*$ and report $\succ^* \circ(r_\ell, 1)$. Since this holds for all ℓ, it follows that o allocates every agent $a_1, \ldots, a_K$ an item indexed larger than K.

We now show how to obtain the S_ℓ sets. The base case is $S_1 = \{K+1, \ldots, n\}$, which satisfies the stated property. Given a set S_ℓ at the end of stage $\ell < K$ we now show how to construct the set $S_{\ell+1} \subseteq S_\ell$. We construct the following hypergraph H_ℓ. The vertices are the agents in S_ℓ. The hyperedges are subsets of vertices of size at most $(\ell+1)$ constructed as follows. For every ℓ-size subset $\{a_1, \ldots, a_\ell\}$ of S_ℓ, and every $a \in S_\ell$ (which could be the same as one of the a_ts), we add the hyperedge $\{a_1, \ldots, a_\ell, a\}$ if there exist ℓ distinct integers $r_1, \ldots, r_\ell \in [K]$ such that $\mathcal{M}(\succ^{(a_1,r_1),\ldots,(a_\ell,r_\ell)})$ allocates agent a an item with index at most K. Note that the number of hyperedges is at most $|S_\ell|^\ell \cdot K^{\ell+1}$ since there are $|S_\ell|$ choices for each a_t, and K choices for each r_t, and once these are fixed, there at most K choices for a.

Call a subset $U \subseteq S_\ell$ *independent* if no hyperedge is completely contained in it. Observe that U is a valid input to stage $(\ell+1)$ if $|U| \geq \ell+1$: consider any $t < \ell+1$ agents $a_1, \ldots, a_t \in U$ and any distinct integers $r_1, \ldots, r_t \in [K]$. Suppose that $\mathcal{M}(\succ^{(a_1,r_1),\ldots,(a_t,r_t)})$ allocates some agent $a \in U$ an item with index at most K. Then we must have $t = \ell$, otherwise this would contradict the property assumed of S_ℓ, and then $\{a_1, \ldots, a_t, a\}$ would be a hyperedge, contradicting independence of U.

Lemma 4.5 shows that there is an independent set $S_{\ell+1} \subseteq S_\ell$ such that $|S_{\ell+1}| \geq \frac{|S_\ell|^{\frac{1}{\ell+1}}}{K} - 1$. Therefore, if $|S_\ell|^{\frac{1}{\ell+1}} \geq 2K$ (and hence $|S_t|^{\frac{1}{t+1}} \geq 2K$ for all $t \leq \ell$), then

$$|S_{\ell+1}| \geq |S_\ell|^{\frac{1}{\ell+1}}/2K \geq |S_1|^{\frac{1}{(\ell+1)!}}/(2K)^\ell \geq \left(\frac{n}{2}\right)^{\frac{1}{(\ell+1)!}}/(2K)^\ell.$$

If $K \leq \frac{\log\log n}{\log\log\log n} - 2$, then this implies that $\left(\frac{n}{2}\right)^{\frac{1}{K!}}/(2K)^{K-1} \geq 2K$. Hence, $|S_{\ell+1}| \geq 2K$, and moreover, if $\ell+1 < K$, then $|S_{\ell+1}|^{\frac{1}{\ell+2}}/2K \geq \left(\frac{n}{2}\right)^{\frac{1}{(\ell+2)!}}/(2K)^{\ell+1} \geq 2K$. Thus, we obtain that $|S_K| \geq 2K$. $\square$

LEMMA 4.5. *There exists an independent set $S_{\ell+1} \subseteq S_\ell$ of size $|S_{\ell+1}| \geq \frac{|S_\ell|^{\frac{1}{\ell+1}}}{K} - 1$.*

PROOF. Let $\mathcal{N} = |S_\ell|$. Recall the number of hyperedges is at most $\mathcal{N}^\ell K^{\ell+1}$. We first argue that all hyperedges are of size $\ell+1$. Every hyperedge is of size at least ℓ. A size-ℓ hyperedge $\{a_1, \ldots, a_\ell\}$ can only arise, if there are ℓ distinct integers $r_1, \ldots, r_\ell \in [K]$ and some $a \in \{a_1, \ldots, a_\ell\}$, say a_1 for notational convenience such that $\mathcal{M}(\succ^{(a_1,r_1),\ldots,(a_\ell,r_\ell)})$ allots a an item indexed less than K. But the definition of S_ℓ implies that $\mathcal{M}(\succ^{(a_2,r_2),\ldots,(a_\ell,r_\ell)})$ allots a_1 an item with index larger than K. This violates truthfulness, since agent

a_1 has an incentive to misreport $\succ^* \circ(r_1, 1)$ when his true preference is $\succ^*$ and obtain a better item.

Consider sampling each vertex of H_ℓ with probability $p = K^{-1} \cdot \mathcal{N}^{-\left(\frac{\ell}{\ell+1}\right)}$ to get a random subset X. If X contains a hyperedge, then we remove all its vertices from X. The probability that a hyperedge is present in X is at most $p^{\ell+1}$, since all hyperedges are of size $\ell + 1$. Therefore, in expectation, the size of X after removal is at least $p\mathcal{N} - p^{\ell+1}\mathcal{N}^\ell K^{\ell+1} = \frac{\mathcal{N}^{1/\ell+1}}{K} - 1$. $\square$

4.1 A generalization: matroid markets

In this generalization of matching markets, there is a matroid $M_i = (N, \mathcal{I}_i)$ for each item i, and multiple agents may be assigned to item i provided they form an independent set of M_i. Here $\mathcal{I}_i$ is a collection of subsets of N with the following properties: (i) $\emptyset \in \mathcal{I}_i$; for all $A, B \subseteq N$ (ii) if $A \in \mathcal{I}_i$ and $B \subseteq A$, then $B \in \mathcal{I}_i$; (iii) if $A, B \in \mathcal{I}_i$ and $|A| > |B|$, then there exists some $j \in A \setminus B$ such that $B \cup \{j\} \in \mathcal{I}_i$. Clearly, the lower bounds obtained for matching markets also hold in this setting. Complementing this, we extend MaxMatch to obtain a pseudomonotone 2-rank-approximation algorithm for matroid markets.

THEOREM 4.6. *There is a pseudomonotone* 2*-rank approximation algorithm for matroid markets, and a mechanism that fully LT-implements it.*

PROOF. The algorithm is similar to MaxMatch. Again fix an agent-ordering and an item-ordering. Consider some input $\succ$.

We again proceed in m stages. In stage r, we consider the "top-r" graph $G_r = (N \cup O, E_r)$, where each agent j has edges to his top-r items. Note that every outcome induces a feasible solution to the *matroid-intersection* problem defined by the following two matroids on the universe E_r. One is M_A, which is the direct sum of the M_i matroids for all $i \in O$, i.e., a set $I \subseteq N \times O$ is independent if $\{j : (j, i) \in I\} \in \mathcal{I}_i$ for all $i \in O$. The second is the partition matroid $M_B(r)$ encoding that at most one edge of E_r is incident to each item j. Then every outcome induces a set that is independent in both M_A and $M_B(r)$, and $\text{maxrank}_r(\succ)$ is the size of the largest common independent set.

Let M consist of the edges denoting the current (i.e., at the start of stage r) assignment of items to agents. Our algorithm will maintain the invariant that at the end of stage r, M is a *maximal* set that is independent in both M_A and $M_B(r)$. The rank-approximation factor of 2 follows then from the well-known fact that every maximal common independent set of two matroids has size at least half the size of maximum-cardinality common independent set; Claim 4.7 gives a self-contained proof.

Let $\Gamma^r(u)$ denote the neighbors of node u in G_r, and $\Gamma^r_M(u) := \{v : (u, v) \in M\}$. Note that if M is a maximal common independent set in G_{r-1}, then for every agent j that is not assigned an item in M, among j's top-r items his r-th ranked item is the *only* item to which j can be possibly assigned while preserving independence in the item's matroid.

We consider each item i and augment $\Gamma^r_M(i)$, the current set of agents assigned to item i, to a maximal subset $J_i \subseteq \Gamma^r(i)$ that is independent in M_i: we initialize J_i to $\Gamma^r_M(i)$. Next, we consider agents in $\Gamma^r(i) \setminus \Gamma^r_M(i)$ according to the fixed agent-ordering and add agent j to J_i if this maintains independence in M_i. Maximality of J_i follows from the matroid property. (In fact J_i is a maximum-size independent subset of $\Gamma^r(i)$.) Finally, we update M to reflect the new assignments in stage r.

The fact that M is a maximal common independent set of M_A and $M_B(r)$ is immediate: if some edge (j, i) can be added to M while preserving independence in M_A and $M_B(r)$, then j was unassigned at the start of stage r and when we considered item i, j could (and would) have been added to J_i in the iteration when j was considered.

We have already argued that the above algorithm is a 2-rank-approximation. Pseudomonotonicity of the above algorithm follows from exactly the same arguments as in Theorem 4.1. $\square$

CLAIM 4.7. *Let $M_1(U, \mathcal{I}_1)$, $M_2 = (U, \mathcal{I}_2)$ be two matroids. Let $S \subseteq U$ be an inclusion-wise maximal set that is independent in both M_1 and M_2. Let A be a maximum-cardinality set that is independent in both M_1 and M_2. Then $|S| \geq |A|/2$*

PROOF. Suppose $|S| < |A|/2$. Let $T_1 = \{e \in A : S \cup \{e\} \in \mathcal{I}_1\}$. Since $A \in \mathcal{I}_1$, by the matroid exchange property, we have $|T_1| \geq |A| - |S| > |A|/2$. Similarly, if $T_2 = \{e \in A : S \cup \{e\} \in \mathcal{I}_2\}$, then we have $|T_2| > |A|/2$. But since $T_1, T_2 \subseteq A$, this means that $T_1 \cap T_2 \neq \emptyset$, and so if $e \in T_1 \cap T_2$, then e can be added to S while maintaining independence in both M_1 and M_2. This contradicts the maximality of S. $\square$

5. SCHEDULING MARKETS

Recall that here the agents are n jobs that need to be assigned on m machines. Each job j has a private strict total ordering over the machines, and a publicly-known processing time p_{ij} on machine i. An outcome is a partial assignment of jobs to machines, also called a schedule, that has makespan at most a given value T. An agent prefers outcome o to outcome o' if he prefers his assigned machine in o to that in o'.

We obtain nearly tight results for scheduling markets. Say that an algorithm is an (α, β)-approximation if it always returns a schedule with rank-approximation factor α and makespan at most βT. For parallel machines ($p_{ij} = p_j$ for all i, j), we give an $\big(O(\log n), O(1)\big)$-approximation, fully lex-truthfully (LT) implementable algorithm (Theorem 5.2). We show that this bound is *tight* by proving an *algorithmic lower bound* showing that every (α, β)-approximation algorithm for parallel machines must have $\alpha = \Omega(\max\{\log m, \log n\}/\beta)$ (Theorem 5.3). For the setting of general unrelated machines, we devise an $\big(O(\log n), O(1)\big)$-approximation algorithm (Theorem 5.4), however we do not know how to achieve this via a fully LT-implementable algorithm. We leave this as an intriguing open question.

Let N denote the set of jobs. For $S \subseteq N$, let $\succ_S$ denote the restriction of $\succ$ to jobs in S, and $\text{maxrank}_r(\succ_S)$ denote the maximum number of jobs from S that can be assigned to one of their top-r machines with makespan at most T. Observe that $\text{maxrank}_r(\succ_{S \cup T}) \leq \text{maxrank}_r(\succ_S) + \text{maxrank}_r(\succ_T)$.

Parallel machines.

Our results rely on a bucketing argument coupled with Theorem 4.6 for matroid-markets and some insights from the matroid-intersection problem. We divide the set N of jobs into $k = O(\log n)$ disjoint classes $N_0, N_1, \ldots, N_k$ such that jobs in each class have roughly the same processing time. Set $N_0 := \{j : p_j \leq \frac{T}{n}\}$, and $N_\ell := \{j : 2^{\ell-1} \cdot \frac{T}{n} < p_j \leq 2^\ell \cdot \frac{T}{n}\}$

for $\ell = 1, \ldots, k := \lceil \log_2 n \rceil$. Note that if $j \notin \bigcup_{\ell=0}^{k} N_\ell$, then $p_j > T$, so j cannot be assigned to any machine in any outcome and is not counted in $\text{maxrank}_r(\succ)$ for any position r. We assume for notational convenience that N does not contain any such job in the sequel. It will be convenient to ensure that $|N_0| \geq 1$. So we remove some fixed job a from the N_ℓ set containing it and add it to N_0.

Obtaining a good rank-approximation for a class N_ℓ, where $\ell \geq 1$, with makespan $O(T)$ amounts to a matroid-market problem (in fact, a *b-matching* problem) since the makespan bound can be encoded by the constraint that at most $\frac{n}{2^{\ell-1}}$ jobs are assigned to each machine. Any feasible schedule for N_ℓ yields a feasible allocation for the corresponding matroid-market problem. So Theorem 4.6 yields a pseudomonotone $(2, 2)$-approximation algorithm f_ℓ for class N_ℓ, and a mechanism $\mathcal{M}_\ell^\varepsilon$ that ε-implements it, for all $\varepsilon > 0$.

Theorem 5.1. *One can obtain a deterministic fully LT-implementable $\big(O(1), O(\log n)\big)$-approximation algorithm for parallel-machine markets.*

Proof. On input $\succ$, we output the schedule obtained by concatenating the schedule where all jobs in N_0 are assigned to their top machine, and all the $f_\ell(\succ_{N_\ell})$ schedules. Note that the N_0-schedule has makespan at most $2T$. The resulting schedule, denoted $f(\succ)$, has makespan $O(T \log n)$ and rank-approximation factor 2 (since $\text{maxrank}_r(\succ_N) \leq \sum_{\ell=0}^{k} \text{maxrank}_r(\succ_{N_\ell})$). Fix $\varepsilon > 0$. The jobs in N_0 clearly have no incentive to lie. It is easy to see then that f is ε-LT implemented by the mechanism that outputs the N_0-schedule concatenated with the (random) schedules output by the $\mathcal{M}_\ell^\varepsilon$ mechanisms, where we couple the random choices of all the $\mathcal{M}_\ell^\varepsilon$ mechanisms (i.e., their decisions are based on the outcomes of the same random coins) so that $\mathbf{Pr}[\exists\ \ell : \mathcal{M}_\ell^\varepsilon(\succ_{N_\ell}) \neq f_\ell(\succ_{N_\ell})] \leq \varepsilon$. $\square$

Theorem 5.2. *There is a randomized fully LT-implementable $\big(O(\log n), O(1)\big)$-approximation algorithm for parallel-machine markets, where the rank-approximation and makespan bounds hold with probability 1.*

Proof. Consider an input $\succ$. As before, we assign all jobs in N_0 to their top machine. Note that simply picking a class N_ℓ with probability $\frac{1}{k}$ and outputting the concatenation of the N_0-schedule and $f_\ell(\succ_{N_\ell})$ is not enough since this only yields $O(k)$ rank approximation in expectation. Instead, we build upon the above ideas and leverage some results about the matroid-intersection problem.

Consider the following bipartite graph representing the concatenation σ of all the $f_\ell(\succ_{N_\ell})$ schedules. We have a node for every machine, and every job not in N_0, and an edge (i, j) if j is assigned to machine i in schedule σ. Now set $x_{ij} = \frac{1}{k}$ for every edge (i, j). Define $A_{i,\ell} := \big\lceil \frac{n}{2^{\ell-1}k} \big\rceil$ for all i, ℓ and $B_r := \big\lfloor \text{rank}_r(\sigma; \succ_{N \setminus N_0})/k \big\rfloor$ for all r. Consider the following polytope:

$$\mathcal{P} := \Big\{ y \in \mathbb{R}^{[m] \times (N \setminus N_0)} : \sum_{j \in N_\ell} y_{ij} \leq A_{i,\ell} \quad \forall i \in [m], \ell \in [k], \\ \sum_{j : \text{pos}(\succ_j, \sigma(j)) \leq r} y_{ij} \geq B_r \quad \forall r \in [m], \\ 0 \leq y_{ij} \leq 1 \quad \forall i \in [m],\ j \notin N_0 \Big\}.$$

We claim that $\mathcal{P}$ has integral extreme points. Any extreme point of $\mathcal{P}$ is defined by a linearly independent system of tight constraints comprising some $\sum_{j \in N_\ell} y_{ij} = A_{i,\ell}$ equalities whose supports are disjoint, and some $\sum_{j : \text{pos}(\succ_j, \sigma(j)) \leq r} y_{ij} = B_r$, $y_{ij} = 1$ equalities whose supports form a laminar family. The constraint matrix of such a system thus corresponds to equations coming from two laminar set systems; such a matrix is known to be totally unimodular (TU) (see, e.g., [26]), and hence a solution to this system is integral.

Note that $x \in \mathcal{P}$, so it can be expressed as a convex combination of some extreme points of $\mathcal{P}$. Equivalently, x yields a distribution over partial schedules for $N \setminus N_0$. Let Y be a random schedule, or equivalently vector in $\mathbb{R}^{[m] \times (N \setminus N_0)}$, sampled from this distribution. Note that $\mathbf{Pr}[j \text{ is assigned in } Y] = x_{ij} = \frac{1}{k}$ for $j \notin N_0$. The makespan of Y is at most $6T$ with probability 1. This is because $\sum_j p_j Y_{ij} \leq \sum_{\ell=1}^{k} \big(1 + \frac{n}{2^{\ell-1}k}\big) \cdot 2^\ell \cdot \frac{T}{n} \leq 2^{k+1} \cdot \frac{T}{n} + 2T \leq 6T$. Let Π be the (random) schedule obtained by concatenating the N_0-schedule with Y. Then Π has makespan at most $8T$ with probability 1. Also, $\text{rank}_r(\Pi; \succ) \geq |N_0| + B_r$ with probability 1. Now $B_r \geq \big\lfloor \text{maxrank}_r(\succ_{N \setminus N_0})/2k \big\rfloor$. Finally,

$$\begin{aligned} \text{maxrank}_r(\succ) &\leq |N_0| + \text{maxrank}_r(\succ_{N \setminus N_0}) \\ &\leq |N_0| + \max\{2k, 4k \big\lfloor \text{maxrank}_r(\succ_{N \setminus N_0} / 2k) \big\rfloor\} \\ &\leq 4k(|N_0| + B_r), \end{aligned}$$

where the latter inequality follows since $|N_0| \geq 1$. Thus, the randomized algorithm f that outputs the random schedule Π is an $\big(O(\log n), O(1)\big)$-approximation with probability 1.

We now proceed as in the proof of Theorems 3.3 and 3.4 to devise a mechanism $\mathcal{M}$ that fully LT-implements f. Fix $\varepsilon > 0$, and $\varepsilon_1 > \ldots > \varepsilon_m$ such that $\sum_{r=1}^{m} \varepsilon_r = \varepsilon$. Consider input $\succ$. Let $Y^\succ$ be the random schedule for $N \setminus N_0$ for input $\succ_{N \setminus N_0}$ as obtained above. Mechanism $\mathcal{M}$ always assigns jobs in N_0 to their top machines. For jobs in $N \setminus N_0$, it returns schedule $Y^\succ$ with probability $1 - \varepsilon$. For each $j \notin N_0$ and $r \in [m]$, with probability $\frac{\varepsilon_r}{n}$, it returns the schedule where j is assigned to its r-th ranked machine $\text{alt}(\succ_j, r)$, and all other jobs are unassigned. Clearly, $\mathcal{M}(\succ) = f(\succ)$ with probability at least $1 - \varepsilon$.

Jobs in N_0 do not benefit by lying. Consider a job $j \in N_\ell$, where $\ell \geq 1$. Let $\succ' = (\succ_j, \succ_{-j})$, where $\succ'_j \neq \succ_j$. Let $x_{ij} = x_{ij}(\succ)$ and $x'_{ij} = x_{ij}(\succ')$ denote the probabilities that j is assigned to i under the random schedules $Y = Y^\succ$ and $Y' = Y^{\succ'}$ respectively. Then,

$$\begin{aligned} \Delta_{ij} &:= \mathbf{Pr}[j \text{ assigned to } i \text{ in } \mathcal{M}(\succ)] - \mathbf{Pr}[j \text{ assigned to } i \text{ in } \mathcal{M}(\succ')] \\ &= (1 - \varepsilon)(x_{ij} - x'_{ij}) + \frac{1}{n} \cdot \big(\varepsilon_{\text{pos}(\succ_j, i)} - \varepsilon_{\text{pos}(\succ'_j, i)}\big). \end{aligned}$$

Considering machines in the preference order of $\succ_j$, let $\hat{i}$ be the first machine such that $\text{pos}(\succ_j, \hat{i}) \neq \text{pos}(\succ'_j, \hat{i})$. Then $\text{pos}(\succ_j, \hat{i}) < \text{pos}(\succ'_j, \hat{i})$. If $x'_{ij} = 0$ for all $i \succeq_j \hat{i}$, then $\Delta_{ij} \geq 0$ for all $i \succeq_j \hat{i}$, and $\Delta_{\hat{i}j} > 0$, so we are done. Otherwise, j is assigned to some machine $i' \succeq_j \hat{i}$ in $f_\ell(\succ'_{N_\ell})$. Since all machines $i \succ_j i'$ have $\text{pos}(\succ_j, i) = \text{pos}(\succ'_j, i)$ and f_ℓ is pseudomonotone, it must be that j is assigned to $i'' \succeq_j i'$ in $f_\ell(\succ_{N_\ell})$. So $x_{ij} = x'_{ij}$, and hence, $\Delta_{ij} = 0$, for all $i \succ_j i''$. If $i'' \neq i'$, then $\Delta_{i''j} > 0$, otherwise $\Delta_{ij} = 0$ for all $i \succ_j \hat{i}$ and $\Delta_{\hat{i}j} > 0$. Thus, $\mathcal{M}$ is lex-truthful. $\square$

Theorem 5.3. *There exists an instance of a parallel-machine market where any schedule with βT makespan has rank-approximation factor $\Omega(\max\{\log m, \log n\}/\beta)$.*

PROOF. We create an instance with $n = O(m \ln m)$ jobs as follows. We create a set $A^{(1)}$ of m jobs of size (i.e., p_j) T partitioned into $A^{(1)}_1, \ldots, A^{(1)}_m$, where each $A^{(1)}_i$ consists of a single job whose first preference is machine i. We create a set $A^{(2)}$ of $2(m-1)$ jobs of size $\frac{T}{2}$ partitioned into $A^{(2)}_2, \ldots, A^{(2)}_m$, all of which have machine 1 as their first preference. Each set $A^{(2)}_i$ has two jobs, both having machine i as their 2nd preference. In general for $i < k$, we have a set $A^{(i)}$ of $2^{i-1}(m-i+1)$ jobs of size $\frac{T}{2^{i-1}}$ partitioned into $A^{(i)}_i, \ldots, A^{(i)}_m$, all of which have machine r as their r-th preference for $r = 1, \ldots, i-1$. Each set $A^{(i)}_\ell$ has 2^{i-1} jobs, all of which have machine ℓ as their i-th preference. Finally, we have a set $A^{(k)}$ of $2^k m$ jobs of size $\frac{T}{2^k}$ partitioned into $A^{(k)}_k, \ldots, A^{(k)}_m$, all having machine r as their r-th preference for $r = 1, \ldots, k-1$. Each set $A^{(k)}_\ell$ has at least 2^k jobs, all of which have machine ℓ as their k-th preference. The remaining preferences of the jobs play no role, and may be set arbitrarily. Let $\succ$ be the resulting preference profile.

For $r \in [k]$, we have $\operatorname{maxrank}_r(\succ) \geq 2^{r-1}(m-r+1) + 2^k(r-1) \geq 2^{r-1}m$ obtained by assigning all jobs in $A^{(r)}_\ell$ to machine ℓ for $\ell = r, \ldots, m$, and any $2^k(r-1)$ jobs from $A^{(k)}$ to machines $1, \ldots, r-1$. Suppose we have a schedule σ with makespan βT that achieves α rank approximation. Then, $\operatorname{rank}_r(\sigma; \succ) \geq \frac{2^{r-1}m}{\alpha}$ for all $r = 1, \ldots, k$. Let s_r be the number of jobs assigned to their r-th ranked machine in σ, and t_r be the number of jobs of size at least $\frac{T}{2^{r-1}}$ assigned to their r-th ranked machine in σ. Observe that $t_r \geq s_r - \beta 2^k$ since the jobs counted in s_r but not in t_r lie in $\bigcup_{\ell=r+1}^k A^{(\ell)}$, all of which have machine r as their r-th ranked machine; at most $\beta 2^k$ such jobs can be accommodated within makespan βT. Now $\beta m T$ is at least the total size of all jobs scheduled by σ, which is at least $\sum_{r=1}^k (s_r - \beta 2^k) \cdot \frac{T}{2^{r-1}} \geq \sum_{r=1}^k s_r \cdot \frac{T}{2^{r-1}} - \beta 2^{k+1} T$. So

$$
\begin{aligned}
\beta(mT + 2^{k+1}T) &\geq \frac{1}{2} \sum_{r=1}^k s_r \sum_{\ell=r}^k \frac{T}{2^{\ell-1}} \\
&= \frac{1}{2} \sum_{\ell=1}^k \frac{T}{2^{\ell-1}} \sum_{r=1}^{\ell} s_r = \frac{1}{2} \sum_{\ell=1}^k \frac{T}{2^{\ell-1}} \cdot \operatorname{rank}_\ell(\sigma; \succ) \\
&\geq \frac{1}{2} \sum_{\ell=1}^k \frac{T}{2^{\ell-1}} \cdot \frac{2^{\ell-1}m}{\alpha}.
\end{aligned}
$$

Taking $k = \log_2 m$, this gives $3\beta m T \geq \frac{kmT}{2\alpha}$, so $\alpha \geq \frac{k}{6\beta} = \Omega(\log m/\beta)$. Also, the number of jobs is at most $k \cdot 2^k = O(m \log m)$, so α is also $\Omega(\log n/\beta)$. $\square$

Unrelated machines.

We obtain an $\big(O(\log n), O(1)\big)$ approximation for the general setting of unrelated machines.

THEOREM 5.4. *There is a deterministic $\big(O(\log n), O(1)\big)$ approximation algorithm for scheduling markets.*

PROOF. We will need Lemma 5.5 stated below. Fix an input $\succ$. We use a different kind of bucketing argument where we group ranks that have roughly the same value of $\operatorname{maxrank}_r(\succ)$. For $r \in [m]$, let σ^r be the schedule given by Lemma 5.5 that yields a 2-approximation to $\operatorname{maxrank}_r(\succ)$, N_r be the set of jobs assigned by σ_r, and $n_r = |N_r|$. We may assume that $n_1 \leq n_2 \leq \ldots n_m$. Define $n_0 = 0$. If $n_m = 0$, then $\operatorname{maxrank}_r(\succ) = 0$ for all $r \in [m]$, and we return the null assignment. So assume otherwise in the sequel.

Define $r_0 := 0 < r_1 < r_2 < \ldots < r_k < r_{k+1} = m+1$ as follows: r_ℓ is the smallest r such that $n_r > 4n_{r_{\ell-1}}$ for $\ell = 1, \ldots, k$, and $n_m \leq 4n_{r_k}$. Thus, $k \leq \lceil \log_4 n \rceil$ and $n_{r_\ell} \leq n_r \leq 4n_{r_\ell}$ for all $r \in [r_\ell, r_{\ell+1})$ and all $\ell = 0, \ldots, k$. For $\ell \in [k]$, define $S_{r_\ell} := N_{r_\ell} \setminus (\bigcup_{q=1}^{\ell-1} N_{r_q})$; note that $|S_{r_\ell}| \geq \frac{2n_{r_\ell}}{3}$.

If $n_{r_k} < 2k$ for all r, we simply return the assignment σ^{r_1}. Clearly, this yields a $2k$ rank approximation. Otherwise, let q be the smallest index ℓ such that $n_{r_\ell} \geq 2k$. Let $S = \bigcup_{\ell=q}^k S_{r_\ell}$. Let σ be the schedule for S, where each job $j \in S_{r_\ell}$ is assigned to the machine $\sigma^{r_\ell}(j)$, for $\ell = q, \ldots, k$. Let $L_i := |\{j : \sigma(j) = i\}|$. Consider the following bipartite graph, which is similar to the bipartite graph constructed in the GAP-rounding algorithm [10]. We have a node for every job in S, and a node (i, c) for every machine i and $c = 1, \ldots, \left\lceil \frac{L_i}{k-q+1} \right\rceil$. We sort the jobs assigned to i in σ in non-increasing p_{ij} order (breaking ties arbitrarily), and create an edge $\big((i,c), j\big)$ if $\sigma(j) = i$ and its position in this ordering lies in $\{(c-1)(k-q+1)+1, \ldots, c(k-q+1)\}$. Let E be the edge-set of this bipartite graph. Consider the following polytope:

$$
\begin{aligned}
\mathcal{Q} := \Big\{ y \in \mathbb{R}^E : \sum_{j:((i,c),j) \in E} & y_{(i,c),j} \leq 1 \\
& \forall i \in [m], c = 1, \ldots, \left\lceil \tfrac{L_i}{k-q+1} \right\rceil, \\
\sum_{((i,c),j) \in E : j \in S_{r_\ell}} & y_{(i,c),j} \geq \left\lfloor \frac{|S_{r_\ell}|}{k-q+1} \right\rfloor \\
& \forall \ell = q, \ldots, k, \\
& 0 \leq y_{(i,c),j} \leq 1 \quad \forall \big((i,c), j\big) \in E \Big\}.
\end{aligned}
$$

As with the polytope $\mathcal{P}$ (see Theorem 5.2), the constraint-matrix defining an extreme point of $\mathcal{Q}$ corresponds to equations coming from two laminar systems, which is TU, so $\mathcal{Q}$ has integral extreme points. Setting $x_{(i,c),j} = \frac{1}{k-q+1}$ for every edge $\big((i,c), j\big)$, note that $x \in \mathcal{Q}$. So we can find an integral $y \in \mathcal{Q}$, which we interchangeably view as a partial assignment of S. We return the schedule π obtained by concatenating σ^{r_1} with this assignment y.

The schedule σ^{r_1} has makespan at most T. By the standard GAP-rounding proof in [10], the makespan of y is at most

$$
T + \tfrac{1}{k-q+1} \cdot \sum_{j:\sigma(j)=i} p_{ij} = T + \tfrac{1}{k-q+1} \cdot \sum_{\ell=q}^k \sum_{j \in S_{r_\ell} : \sigma^{r_\ell}(j) = i} p_{ij} \leq 2T.
$$

So π has makespan at most $3T$. Fix some rank r. If $r < r_1$, then $\operatorname{maxrank}_r(\succ) = 0$. If $r_1 \leq r < r_q$, we have $\operatorname{rank}_r(\pi; \succ) \geq n_{r_1} \geq 1 > n_r/2k \geq \operatorname{maxrank}_r(\succ)/4k$. Otherwise, suppose $r \in [r_\ell, r_{\ell+1})$, where $\ell \geq q$. Then $\operatorname{rank}_r(\pi; \succ) \geq \left\lfloor \frac{|S_{r_\ell}|}{k} \right\rfloor \geq \left\lfloor \frac{2n_{r_\ell}}{3k} \right\rfloor \geq \frac{n_{r_\ell}}{3k}$, where the last inequality follows since $n_{r_\ell} \geq n_{r_q} \geq 2k$, and $\frac{n_{r_\ell}}{3k} \geq \frac{n_r}{12k} \geq \frac{\operatorname{maxrank}_r(\succ)}{24k}$. So π has $O(k)$ rank approximation. $\square$

LEMMA 5.5. *For any preference-profile $\succ$, any set $S \subseteq N$, and any rank r, one can efficiently compute a 2-approximation to* $\operatorname{maxrank}_r(\succ_S)$.

PROOF. Shmoys and Tardos [10] proved the following result about GAP. Let $\{c_{ij}\}$ be "assignment costs" for assigning jobs to machines, *which could also be negative.* Consider the following LP, where i indexes the machines and j indexes the jobs.

$$\min \quad \sum_{i,j} c_{ij}x_{ij} \tag{P}$$

$$\text{s.t.} \quad \sum_i x_{ij} = 1 \qquad \forall j \tag{1}$$

$$\sum_j t_{ij}x_{ij} \le L_i \qquad \forall i$$

$$x_{ij} \ge 0 \qquad \forall i, j$$

$$x_{ij} = 0 \qquad \forall i, j \text{ s.t. } t_{ij} > L_i.$$

[10] showed that a fractional solution x to (P) can be rounded to an integer solution $\tilde{x}$ of cost at most the cost of x such that the total load $\sum_j t_{ij}\tilde{x}_{ij}$ on each machine i is at most $2L_i$. Examining their rounding algorithm more closely, one can infer that the total load on each machine i under $\tilde{x}$ is at most L_i if $\sum_j x_{ij} \le 1$, and at most $L_i + \max_{j:x_{ij}>0} t_{ij}$ otherwise.

We apply this result to our problem of approximating $\text{maxrank}_r(\succ_S)$ as follows. The set of jobs is S. Our problem is a prize-collecting problem, which we can reduce to GAP by creating a "machine" I_j for every job $j \in S$ and setting $t_{I_j k} = 0$ if $k = j$ and ∞ otherwise. There is no makespan bound for these I_j machines. For every (regular) machine i and job j, we set $t_{ij} = p_{ij}$ if $\text{pos}(\succ_j, i) \le r$ and ∞ otherwise; the makespan bound for i is T. Finally, our objective is to maximize the number of jobs assigned to the regular machines (with $t_{ij} < \infty$). In terms of the LP (P), constraint (1) now reads $\sum_i x_{ij} + z_j = 1$ for every $j \in S$, where z_j indicates if j is assigned to I_j, and the objective function is to maximize $\sum_i \sum_{j\in S} x_{ij}$.

Applying the GAP-rounding algorithm, we obtain an assignment $\tilde{x}$ with makespan at most $2T$ such that $\text{rank}_r(\tilde{x}; \succ_S) = \text{maxrank}_r(\succ_S)$. To turn this into a feasible schedule with makespan T, we leverage the stronger property of the rounding algorithm mentioned above. If the load on machine i under $\tilde{x}$ is more than T, then we know that i has at least two jobs assigned to it, and there is a job assigned to i whose removal decreases the load on i to at most T. We simply remove this job from every overloaded machine i. This reduces the number of jobs assigned to an overloaded machine i by a factor of at most 2 (since $\sum_j \tilde{x}_{ij} \ge 2$), so now we obtain a schedule with makespan T where the number of jobs assigned (to one of their top-r machines) is at least $\text{maxrank}_r(\succ_S)/2$. □

6. MECHANISMS FOR GENERAL ORDINAL SETTINGS

In this section, we evaluate the strength and flexibility provided by the notions of rank approximation and lex-truthfulness in general ordinal settings. We devise an $O(\log n)$-rank-approximation randomized mechanism, and show that this guarantee is tight for randomized mechanisms (Theorems 6.2 and 6.3). We also observe that deterministic mechanisms cannot in general achieve good rank approximation. Next, we consider lex-truthfulness and justify our earlier remark that lex-truthfulness allows one to circumvent Gibbard's impossibility result. We describe a rich class of pseudomonotone SCFs called *top-choice SCFs*, which thus lead to (non-unilateral, non-duple) LT mechanisms.

Rank approximation.

It is easy to see that any deterministic dictatorial SCF has rank approximation (at most) n. Also, the *plurality scoring rule* f^{Pl}, which returns the outcome that maximizes the number of agents who have it as their top choice, has $\text{rank}_1\big(f^{\mathsf{Pl}}(\succ); \succ\big) \ge \frac{n}{m}$, so its rank-approximation factor is at most m. It is not hard to prove a matching lower bound for deterministic mechanisms.

THEOREM 6.1. *No deterministic mechanism can have rank approximation better factor than* $\min\{n, m-1\}$ *in general ordinal settings.*

PROOF. Consider a preference profile with n agents and $n+1$ outcomes, where the top choices of all agents are the distinct outcomes $\{1, \ldots, n\}$, while the second choice of all agents is $n+1$. □

Randomization leads to an exponential improvement (but no more), but we do not know how to achieve this in a lex-truthful manner.

THEOREM 6.2. *There is a randomized* $O(\log n)$*-rank approximation mechanism for general ordinal settings.*

PROOF. We first describe the mechanism, and then analyze its rank approximation. Fix a preference profile $\succ$. For brevity, let $n_r = \text{maxrank}_r(\succ)$. Let o^*_r be an outcome with $\text{rank}_r(o^*_r; \succ) = n_r$. We use a bucketing argument where we group ranks that have roughly the same n_r value. Define $r_1 := 1 < r_2 < \ldots < r_k < r_{k+1} := m+1$ be such that $n_{r_\ell} \le n_r \le 2n_{r_\ell} \quad \forall r \in [r_\ell, r_{\ell+1}) \cap \mathbb{Z}$, for all $\ell = 1, \ldots, k$ Observe that $k \le \lceil \log_2 n \rceil$. The mechanism chooses an index $\ell \in [k]$ uniformly at random, and outputs $o^*_{r_\ell}$.

To argue about the rank approximation, consider any rank r. Suppose $r \in [r_\ell, r_{\ell+1})$. If we choose index ℓ, which happens with probability $1/k$, then at least $\text{rank}_r(o^*_{r_\ell}; \succ) \ge \text{rank}_{r_\ell}(o^*_{r_\ell}; \succ) = n_{r_\ell} \ge \frac{n_r}{2}$ agents are allotted a top-r item. So $\mathbf{E}[\text{rank}_r(\mathcal{M}(\succ); \succ)] \ge \frac{n_r}{2k}$. □

THEOREM 6.3. *Every randomized mechanism has rank approximation factor* $\Omega(\log n)$*.*

PROOF. Fix a parameter k. We construct an instance with $n = 2^{k+1} - 2$ agents and $m = (k-1) \cdot (2^{k+1} - 2) + k$ outcomes. The agents are divided into k groups $A_1, \ldots, A_k$, where $|A_\ell| = 2^\ell$. There are k special outcomes $\{o_1, \ldots, o_k\}$. The remaining $m-k$ outcomes are partitioned into n groups $O_1, \ldots, O_n$, each having $k-1$ outcomes. We now describe the preference lists. For every agent j in group A_ℓ, outcome o_ℓ is their ℓ-th ranked outcome, and the outcomes in O_j occupy the other positions in $[k] \setminus \{\ell\}$; the exact positions of these outcomes are irrelevant. The outcomes in positions $r \ge k+1$ are also immaterial. Thus, the top-k outcome sets of agents j and j' are: disjoint if they are from different groups, and have exactly one outcome, o_ℓ, in common, at the ℓ-th position, if they both belong to group A_ℓ. Let $\succ$ denote this input.

Observe that $\text{maxrank}_r(\succ) = 2^r$ for all $r \in [k]$, and the outcome achieving this is o_r. Furthermore, $\text{rank}_r(o)$ is 2^ℓ if $o = o_\ell$ for $\ell \in [r]$, and is at most 1 otherwise.

Now consider a randomized mechanism that attains rank approximation α. Let p_ℓ be the probability with which it returns the outcome o_ℓ. Let q be the probability with which it returns an outcome in $\bigcup_{j=1}^n O_j$. Then, by the definition of rank approximation we have $q+\sum_{\ell>r} p_r+\sum_{\ell\le r} p_\ell\cdot 2^\ell \ge \alpha\cdot 2^r$ for all $r\in[k]$. Dividing this inequality by 2^r and summing over all $r=1,\dots,k$, we obtain that $k\alpha \le q\cdot\sum_{r=1}^k\left(\frac{1}{2^r}\right)+\sum_{\ell=1}^k p_\ell\cdot\left(\sum_{r<\ell}\frac{1}{2^r}+\sum_{r\ge\ell}\frac{2^\ell}{2^r}\right)\le q\cdot 1+\sum_{\ell=1}^k p_\ell\cdot 3\le 3$. Hence, $\alpha\le\frac{3}{k}$. $\square$

Lex-truthful mechanisms.

Consider any SCF of the form $f(\succ)=g\big(\{\mathrm{alt}(\succ_j,1)\}_{j=1}^n\big)$, where $g: O^n\mapsto O$ has the following property: for all $o_{-j}=(o_1,\dots,o_{j-1},o_{j+1},\dots,o_n)\in O^{n-1}$ and all $o\in O$, if $g(o,o_{-j})=o'$ then $g(o',o_{-j})=o'$. We call such an SCF a *top-choice SCF* since it only looks at the top choices of the players. It is not hard to see that the plurality scoring rule f^{Pl} mentioned earlier (with a fixed tie-breaking rule for outcomes) is an example of such an SCF. We show that *any* top-choice SCF is pseudomonotone, and so by Theorem 3.3 is fully LT-implementable.

THEOREM 6.4. *Every top-choice SCF is pseudomonotone, and hence is fully LT-implementable.*

PROOF. Let f be a top-choice SCF defined by $g: O^n\mapsto O$ having the required property. Consider an agent j, and $\succ=(\succ_j,\succ_{-j})$, $\succ'=(\succ'_j,\succ_{-j})$. Let $o=\mathrm{alt}(\succ_j,1)$. If $f(\succ)=o$ or $f(\succ)=f(\succ')$, then we are done. Otherwise, since $f(\succ)\ne o$, we also have $f(\succ')\ne o$ due to the property of g, and also $\mathrm{pos}(\succ'_j,o)>1$ (otherwise $f(\succ)=f(\succ')$), and so the pseudomonotonicity condition (Definition 5) is satisfied. $\square$

7. REFERENCES

[1] A. Abdulkadiroglu and T. Sönmez. Matching markets: Theory and practice. *Prepared for the Econometric Society World Congress, China*, 2010. Available at the first author's website.

[2] D. Abraham, R. Irving, T. Kavitha, and K. Melhorn. Popular matchings. *SIAM Journ. of Computing*, 37:1030 –1045, 2007.

[3] D. Abreu and A. Sen. Virtual implementation in Nash equilibrium. *Econometrica*, 59:997 – 1021, 1991.

[4] Kenneth Arrow. *Social Choice and Individual Values*. Yale University Press, 1951, 2nd ed 1963.

[5] H. Aziz, F. Brandt, and M. Brill. On the tradeoff between economic efficiency and strategyproofness in randomized social choice. In *Proceedings of the 12th international conference on Autonomous agents and multiagent systems (AAMAS)*, 2013.

[6] A. Bhalgat, D. Chakrabarty, and S. Khanna. Social welfare in one sided matching markets without money. In *LNCS Proceedings of Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, 2011.

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

[8] C. Boutilier, I. Caragiannis, S. Haber, T. Lu, A. D. Procaccia, and O. Sheffet. Optimal social choice functions: A utilitarian view. In *Proceedings of the 13th ACM Conference on Electronic Commerce (EC)*, 2012.

[9] D. Chakrabarty and C. Swamy. Welfare maximization and truthfulness in mechanism design with ordinal preferences. *CS arXiv*, `http://arxiv.org/abs/1312.1831`, December 2013.

[10] David B. Shmoys and Eva Tardos. An approximation algorithm for the generalized assignment problem. *Mathematical Programming*, 62:461–474, 1993.

[11] Marie-Jean-Antoine-Nicolas de Caritat de Marquis de Condorcet. Essay on the application of analysis to the probability of majority decisions., 1785.

[12] P. C. Fishburn. Probabilistic social choice based on simple voting comparisons. *Review of Economic Studies*, 51(167):683 – 692, 1984.

[13] P. Gardenfors. Match making: assignments based on bilateral preferences. *Behavioral Sciences*, 20:166 – 173, 1975.

[14] A. Gibbard. Manipulation of voting schemes: a general result. *Econometrica*, 41:587 – 601, 1973.

[15] A. Gibbard. Manipulation of schemes that mix voting with chance. *Econometrica*, 45(3):665 – 681, 1977.

[16] Chien-Chung Huang, Telikepalli Kavitha, Dimitrios Michail, and Meghana Nasre. Bounded unpopularity matchings. In *Proceedings of the 11th Scandinavian Workshop on Algorithm Theory*, 2008.

[17] T. Kavitha, J. Mestre, and M. Nasre. Popular mixed matchings. *Theoretical Computer Science*, Article in Press, 2010.

[18] G. Kreweras. Aggregation of preference orderings. In *Mathematics and Social Sciences I: Proceedings of the seminar on Menthon-Saint-Bernard, France and of Gosing, Austria*, pages 73 – 79, 1965.

[19] Mohammad Mahdian. Random popular matchings. In *Proceedings of the 7th ACM Conference on Electronic commerce*, 2006.

[20] H. Matsushima. A new approach to the implementation problem. *Journal of Economic Theory*, 45:128 – 144, 1988.

[21] A. Postlewaite and D. Schmeidler. Strategic behavior and a notion of ex ante efficiency in a voting model. *Soc. Choice and Welfare*, 3:37 – 49, 1986.

[22] A. D. Procaccia. Can approximation circumvent Gibbard-Satterwaithe? In *Proceedings of 24th AAAI Conference on Artificial Intelligence*, 2010.

[23] A. D. Procaccia and J. S. Rosenschein. The distortion of cardinal preferences in voting. In *Proceedings of 10th CIA, LNCS Series*, volume 4149, pages 317–331. Springer, 2006.

[24] A. E. Roth and M. Sotomayor. *Two-Sided Matching: A Study on Game- Theoretic Modelling*. Cambridge University Press, 1990.

[25] M. Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *J. Econ. Theory*, 10:187 – 217, 1975.

[26] A. Schrijver. *Combinatorial Optimization*. Springer, New York, 2003.

[27] L. Shapley and H. Scarf. On cores and indivisibility. *Journal of Mathematical Economics.*, 1:23 – 28, 1974.

[28] T. Sönmez and U. Ünver. Matching, allocation, and exchange of discrete resources. *Handbook of Social*

Economics, forthcoming, available at second author's website.

[29] H. P. Young. Social choice scoring functions. *SIAM J. Appl. Math.*, 28(4):824–838, 1975.

APPENDIX

A. PROOFS OMITTED FROM SECTION 3

PROOF OF THEOREM 3.2. Clearly $\mathsf{UnivT} \subseteq \mathsf{StrongT}$. If $\mathbf{p}$ stochastically dominates $\mathbf{q}$, then $\mathbf{p}$ lex-dominates $\mathbf{q}$, so $\mathsf{StrongT} \subseteq \mathsf{LexT}$. If $\mathbf{p}$ lex-dominates $\mathbf{q}$, then $\mathbf{q}$ cannot stochastically dominate $\mathbf{p}$, so $\mathsf{LexT} \subseteq \mathsf{WeakT}$. We now prove that the various inclusions are strict.

$\mathsf{UnivT} \subsetneq \mathsf{StrongT}$. Fix a player j. Consider the unilateral mechanism $\mathcal{M}$ that returns one of the top 2 outcomes of j, each with probability $\frac{1}{2}$. $\mathcal{M}$ is clearly strongly truthful. But it is not universally truthful. Consider some input $\succ = (\succ_j, \succ_{-j})$. If $\mathcal{M}$ is a mixture of deterministic truthful mechanisms, then this mixture must assign a probability mass exactly $\frac{1}{2}$ to deterministic truthful mechanisms $\mathcal{M}_1$ satisfying $\mathcal{M}_1(\succ) = o = \mathrm{alt}(\succ_j, 1)$; call these type-1 mechanisms. Similarly, it must assign probability mass exactly $\frac{1}{2}$ to deterministic truthful mechanisms $\mathcal{M}_2$ satisfying $\mathcal{M}_2(\succ) = o' = \mathrm{alt}(\succ_j, 2)$; call these type-2 mechanisms.

For any preference list $\succ_j^*$, a type-2 mechanism cannot return o under the input $(\succ_j^*, \succ_{-j})$ due to truthfulness, otherwise on input $\succ$, j has an incentive to lie in the type-2 mechanism and report $\succ_j^*$. Hence, for any preference list $\succ_j^*$, where o is one of the top two outcomes, *every* type-1 mechanism must return o on input $(\succ_j^*, \succ_{-j})$. A symmetric argument shows that for any preference list $\succ_j^*$ where o' is one of the top two outcomes, every type-2 mechanism must return o' on input $(\succ_j^*, \succ_{-j})$.

Now consider some $\succ_j'$, where the top two outcomes are $o'', \hat{o} \notin \{o, o'\}$. Applying the arguments above we obtain that there are type-3 and type-4 deterministic truthful mechanisms, both of which are assigned probability mass $\frac{1}{2}$ (in the mixture yielding $\mathcal{M}$): the type-3 mechanisms which always return o'' whenever j's preference list has o'' as one of the top two outcomes, and the type-4 mechanisms always return $\hat{o}$ whenever j's preference list has $\hat{o}$ as one of the top two outcomes.

Now some mechanism $\mathcal{M}'$ in the mixture yielding $\mathcal{M}$, must be of multiple types, say type-1 and type-3 for illustration. Then, if $\succ_j''$ has o and o'' as the top two outcomes of j, $\mathcal{M}'$ must return both o and o'' on input $(\succ_j'', \succ_{-j})$, which cannot happen.

$\mathsf{StrongT} \subsetneq \mathsf{LexT}$. Consider a setting with one player and three outcomes: a, b, c. Consider the top-choice SCF f defined by the following function: $g(a) = a$, $g(b) = g(c) = c$, which satisfies the property required for f to be a top-choice SCF. By Theorem 6.4, f is pseudomonotone. Let $\mathcal{M}$ be the LT mechanism that $\frac{1}{3}$-implements f. Let $\succ = (b, a, c)$ denoting that b is top-outcome, and $\succ' = (a, b, c)$. Let $\mathbf{p} = \mathcal{M}(\succ)$ and $\mathbf{q} = \mathcal{M}(\succ')$. Then $\mathbf{p}(b) + \mathbf{p}(a) \leq \frac{1}{3}$ since $f(\succ) = g(b) = c$, but $\mathbf{q}(b) + \mathbf{q}(a) \geq \frac{2}{3}$ since $f(\succ') = g(a) = a$. Thus, $\mathcal{M}$ is not strongly truthful.

$\mathsf{LexT} \subsetneq \mathsf{WeakT}$. Consider a setting with one player and four outcomes: a, b, c, d. Let $\succ^* = (a, b, c, d)$, denoting that a is the top outcome. Define the following randomized mechanism $\mathcal{M}$: $\mathcal{M}(\succ^*)$ returns a with probability $\frac{1}{2}$ and b, c, d with probability $\frac{1}{6}$; on every other input $\succ$, $\mathcal{M}$ returns one of the top three outcomes of $\succ$ with probability $\frac{1}{3}$. $\mathcal{M}$ is weakly truthful, because if $\succ \neq \succ^*$ then $\mathcal{M}$ assigns total probability 1 to the top three outcomes of $\succ$. If $\succ = \succ^*$, then $\mathcal{M}$ assigns probability $\frac{1}{2}$ to the top outcome a under $\succ^*$, whereas for every other input $\mathcal{M}$ assigns probability at most $\frac{1}{3}$ to a.

But $\mathcal{M}$ is not lex-truthful: if $\succ = (a, b, d, c)$, then by reporting (a, b, c, d), the player can increase the probability of his top-outcome a from $\frac{1}{3}$ to $\frac{1}{2}$. $\square$

PROOF OF THEOREM 3.4. We mimic the proof of Theorem 3.3. For all j, and all $r \in [m_j]$, fix some outcome $o_r^j \in O_r^j$ form the indifference class O_r^j of agent j.

We prove part (i) first. Our randomized mechanism $\mathcal{M}$ does the following. On input $\succeq$, it returns $f(\succeq)$ with probability $(1-\varepsilon)$; with probability ε, it picks a random agent a and returns outcome o_r^a with probability $\varepsilon_r^a/\varepsilon$, where $\varepsilon_1^a > \cdots > \varepsilon_{m_a}^a > 0$ are such that $\sum_{r=1}^{m_a} \varepsilon_r^a = \varepsilon$.

Clearly, $\mathcal{M}$ ε-implements f. To prove lex-truthfulness, fix an agent j and consider any $\succeq' = (\succeq_j', \succeq_{-j})$, where $\succeq_j' \neq \succeq_j$. Let $o = f(\succeq)$ and let $o' = f(\succeq')$. Let $O_{t_1}^j$ and $O_{t_2}^j$ be the indifference classes of j containing outcomes o and o' respectively. Also, let $\mathbf{p} = \mathcal{M}(\succeq)$, $\mathbf{q} = \mathcal{M}(\succeq')$.

Considering indifference classes in the preference order of $\succeq_j$, let O_r^j be the first indifference class such that $\mathrm{pos}(\succeq_j, o_r^j) < \mathrm{pos}(\succeq_j', o_r^j)$. Let $o'' = o_r^j$. By pseudomonotonicity of f, we know that $o \succeq_j o'$ or $o'' \succ_j o'$. In the latter case, we have $\mathbf{p}(O_t^j) - \mathbf{q}(O_t^j) \geq 0$ for all t such that $o_t^j \succeq_j o''$, and $\mathbf{p}(O_r^j) - \mathbf{q}(O_r^j) > 0$, so we are done.

If $o \succeq_j o'$, and $O_{t_1}^j = O_{t_2}^j$ or $o'' \succeq_j o$, then the above argument still holds. So suppose $o \succ_j o'$ and $o \succ_j o''$. Then $\mathbf{p}(O_t^j) - \mathbf{q}(O_t^j) \geq 0$ for all t such that $o_t^j \succeq_j o$ and $\mathbf{p}(O_{t_1}^j) - \mathbf{q}(O_{t_1}^j) > 0$, so again we are done.

Now consider part (ii). Let $\mathcal{M}$ be an LT mechanism that ε-LT implements f, where $\varepsilon < \frac{1}{2}$. Suppose for a contradiction, there is some agent j, and $\succeq = (\succeq_j, \succeq_{-j})$ and $\succeq' = (\succ_j', \succeq_{-j})$ such that $o = f(\succ)$ and $o' = f(\succ')$ violate the conditions for pseudomonotonicity. Then we have $o' \succ_j o$ and for every outcome $o'' \succ_j o'$, we have $\mathrm{pos}(\succ_j, o'') = \mathrm{pos}(\succ_j', o'')$. Let $O_{t_1}^j$ and $O_{t_2}^j$ be the indifference classes of j containing outcomes o and o' respectively. Let $\mathbf{p} = \mathcal{M}(\succeq)$ and $\mathbf{q} = \mathcal{M}(\succeq')$.

Since $\mathcal{M}$ ε-LT implements f, we have $\mathbf{p}(O_{t_2}^j) \leq \varepsilon$ and $\mathbf{q}(O_{t_2}^j) \geq \mathbf{q}(o') \geq 1-\varepsilon$, so $\mathbf{p}(O_{t_2}^j) < \mathbf{q}(O_{t_2}^j)$. Let $O_{r_1}^j, \ldots, O_{r_\ell}^j$ be the indifference classes of j that are ranked higher than $O_{t_2}^j$ under $\succeq_j$, ordered so that $o_{r_1}^j \succ_j o_{r_2}^j \succ_j \cdots \succ_j o_{r_\ell}^j$. Since $\mathrm{pos}(\succ_j, o'') = \mathrm{pos}(\succ_j', o'')$ for all $o'' \succ_j o'$, $O_{r_1}^j, \ldots, O_{r_\ell}^j$ are also the indifference class of j that are ranked higher than $O_{t_2}^j$ under $\succeq_j'$, and we have $o_{r_1}^j \succ_j' o_{r_2}^j \succ_j' \cdots \succ_j' o_{r_\ell}^j$. As in the proof of Theorem 3.3, this implies that $\mathbf{p}(O_{r_t}^j) = \mathbf{q}(O_{r_t}^j)$ for all $t = 1, \ldots, \ell$, which contradicts the fact that $\mathcal{M}$ is lex-truthful. $\square$

B. QUALITY OF KNOWN MECHANISMS FOR MATCHING MARKETS

In this section, we investigate the rank approximation and lex-truthfulness of three extensively studied mechanisms for matching markets. These are *random serial dictatorship* mechanism (RSD), Gale's *top-trading-cycle algorithm* (TTCA), and the *probabilistic serial* mechanism (PS).

Random Serial Dictatorship.

Initially all items are marked unallocated. A random permutation of agents is sampled and the agents are considered according to this order. Each agent is allocated his best item among the unallocated items. This item henceforth is marked allocated.

Top Trading Cycle.

This appears in a paper by Shapley and Shubik [27] who attributed it to David Gale and is applicable when the number of items equals the number of agents.

The algorithm starts with an arbitrary assignment σ of agents to items. This assignment, which is called the initial endowment of agents, is independent of the preference orders of the agents. Subsequently, the agents will *trade* among themselves to return the final allocation.

The algorithm then proceeds in rounds. Initially all agents are marked active. In each round, one constructs a directed graph with the active agents as nodes. There is an arc from agent j to agent j', if the item $\sigma(j')$ is the top choice of agent j among the items owned by the active agents, that is, the set $\{\sigma(j) : j \text{ active }\}$. Note that each agent has out-degree exactly 1 (self loops are allowed and counted as both out and in degree). Therefore, there exists at least one directed cycle in the graph. A cycle (self loops are also cycles) is picked arbitrarily. For each arc (j, j') in the cycle, we allocate agents j the item $\sigma(j')$. We mark all agents in this cycle inactive and proceed to the next round. The algorithm stops when all agents are marked inactive.

Probabilistic Serial.

This algorithm is due to Bogomolnaia and Moulin [7]. We first describe the algorithm when the number of agents equals the number of items.

The algorithm first finds a *fractional matching*, that is, x_{ij}'s for items i and agents j such that each $x_{ij} \geq 0$ and $\sum_{i\in I} x_{ij} = 1$ for all agents j, and $\sum_{j\in A} x_{ij} = 1$ for all items i. By the Birkhoff-von Neumann theorem, we can find a distribution on matchings such that the probability agent j is allocated item i is exactly x_{ij}. This is the distribution returned by the algorithm.

The algorithm proceeds in rounds. Initially all x_{ij}'s are 0. For any item i, we denote its capacity as $\sum_{j\in A} x_{ij}$. Any item with capacity strictly < 1 is called unallocated. In each round, every agent points to the best item among the unallocated items. For each unallocated item i we simultaneously raise the x_{ij} for agents j which point to item i at the *same* rate. This continues till some unallocated item's capacity becomes 1. At this point we end the round and proceed to the next round. The algorithm terminates when all items are allocated. Since the procedure maintains that $\sum_{i\in I} x_{ij}$ is same for agents always, at the end we end up with a fractional matching.

A lot of literature exists on all three mechanisms; we point the reader to surveys [28, 1] for a detailed reference. Before stating the rank approximations and lex truthfulness, let us mention some relevant known facts. RSD is strongly truthful (in fact, it is universally truthful). TTCA is the only deterministic algorithm among the three. It is known that for *any* initial endowment, the algorithm is truthful [27]. PS is known to be weakly truthful and not strongly truthful [7]. Bhalgat et al [6] proved that the ordinal welfare factor (cf. Section 1.2) of RS and PSD are 1/2, which is the best possible. The OWF of TTCA is $1/n$ since it is deterministic.

Rank Approximations of RSD, TTCA, and PS.

We show that all three mechanisms have 'bad' rank approximation. Rank approximation of TTCA is at least $(n-1)$, while RSD and PS have rank approximation of $\Omega(\sqrt{n})$. Recall that MaxMatch has rank approximation 2.

We know that TTCA is deterministic and truthful. It is also non-bossy; if an agent changes his preference but still gets the same item, it implies that in the round when he gets allocated an item, the cycle is the same as before, since no other changes preferences. Therefore from Theorem 4.3, we get the rank apporximation is at least $(n-1)$.

Consider an instance $\succ$ with n agents and n items with preference lists as follows. Let $k = \lceil\sqrt{n}\rceil$. Agents $1 \leq i \leq k$ have item i as their top choice. Agents $k+1 \leq i \leq n$ have item n as their top choice. These agents are now grouped into k groups $G_1, \ldots, G_k$, each group containing $n/k - 1$ agents. Agents in group G_ℓ have item ℓ as their second choice. All the other choices of all agents is immaterial and can be assumed to be arbitrary. Observe that $\text{maxrank}_1(\succ) = k + 1$.

Let's first take RSD and calculate the expected number of agents who get their top choice. With $1 - k/n$ probability, an agent $k+1 \leq j \leq n$ shows up as the first agent; he picks item n. No other agent $k+1 \leq j \leq n$ gets his top choice. Henceforth, for any $1 \leq \ell \leq k$, the probability that a guy in G_ℓ shows up before agent ℓ is at least $1 - \frac{2k}{n}$. If that occurs, then agent ℓ doesn't get his top choice. Therefore, the expected number of agents getting their top choice is at most $1 + 2k^2/n + o(n)$. Thus, setting $k = \Theta(\sqrt{n})$, the rank approximation is $\Omega(\sqrt{n})$.

In PS, the calculation is easier. For $1 \leq \ell \leq k$, we get $x_{\ell\ell} = \frac{1}{n-k} + \frac{k}{n}\left(1 - \frac{1}{n-k}\right) = \frac{k-1}{n}$. For agent $k+1 \leq \ell \leq n$, we get $x_{n\ell} = \frac{1}{n-k}$. Therefore, the expected number of agents getting their top choice in PS is precisely $1 + \frac{k(k-1)}{n}$. Setting $k = \Theta(\sqrt{n})$, we get that the rank approximation is $\Omega(\sqrt{n})$.

We do not know if the rank approximation for RSD and PS is $\Theta(\sqrt{n})$ or not.

Lex-Truthfulness of RSD, TTCA, and PS.

TTCA is truthful and RSD is universally truthful. Therefore, they are lex-truthful as well. PS was shown to be weakly truthful by [7]. We show that in fact PS is lex-truthful as well. The proof below is akin to the proof of weak truthfulness in [7] mentioned above; we include it for completeness.

THEOREM B.1. *PS is lex-truthful.*

PROOF. Consider any preference profile $\succ$. By renaming items we may assume $\succ_j = (1, 2, \ldots, n)$ for some agent j. Suppose agent j misreports his preference as $\succ'_j \neq \succ_j$, and let $\succ' := (\succ'_j, \succ_{-j})$. Let k be the first position at which $\succ_j$ and $\succ'_j$ differ. That is, for $r < k$, $\text{alt}(\succ'_j, r) = \text{alt}(\succ_j, r) = r$. Note that j has 'demoted' k in the misreported preference, that is, $\text{pos}(\succ'_j, k) > k$. Let $\mathbf{p}$ and $\mathbf{q}$ be the distributions over items that j obtains on reporting $\succ_j$ and $\succ'_j$ respectively. Let x and x' be the respective fractional matchings.

Observe that since PS has a notion of time (since x_{ij}'s are incremented at a certain rate), we can define $x(t)$ as the assignment at time t. So $x(0) \equiv 0$. Let $t_0 \geq 0$ be the time till which we have $x(t_0) \equiv x'(t_0)$. If t_0 is ill defined, then $x \equiv x'$ and so $\mathbf{p} \equiv \mathbf{q}$ and there's nothing to prove. We must have that till time t_0, agent j points to the same items in both runs, and right after that instant agent j points to different items in the two runs. Say at t_0, agent j pointed to item k in the original run, and k' in the new run. Observe that all items $r < k$ have been completely allocated in both runs since j is pointing to k in the original run. Thus, $\mathbf{p}(r) = \mathbf{q}(r)$ for $r < k$ since $x(t_0) \equiv x'(t_0)$.

We claim $\mathbf{p}(k) > \mathbf{q}(k)$. This will show $\mathbf{p}$ lexicographically dominates $\mathbf{q}$. To do so, we need to introduce some notation. Let t^* and t' be the times at which k is completely allocated in the original and new run respectively. Let t_1 be the time at which j points to k in the new run. Observe $t_0 < t_1 \leq t'$. Also observe that $x_{jk} = t^* - t_0$ and $x'_{jk} = t' - t_1$.

Now, if $t' \leq t^*$, we get $x_{jk} > x'_{jk}$, and we are done. So we may assume $t' > t^*$. For $t \geq t_0$, let $S(t,k)$ and $S'(t,k)$ be the set of agents pointing to item k at time t. Observe that PS satisfies the following monotonicity condition: if an agent points to an item at time t, then he continues to do so till the item is fully allocated. Using this, one can prove the following claim; we defer the proof to the end.

Claim B.2. *For all $t_0 \leq t < t_1$, $|S'(t,k)| \geq |S(t,k)| - 1$, for $t_1 \leq t < t^*$, $|S'(t,k)| \geq |S(t,k)|$, and for $t^* \leq t < t'$, $|S'(t,k)| \geq |S(t^*,k)|$.*

Using the claim, we now show $x_{jk} > x'_{jk}$. Let C denote the capacity of item k at time t_0. We know that $C < 1$. Now, from the run of PS we get

$$\int_{t_0}^{t^*} |S(t,k)| dt \quad = \quad (1 - C) \quad = \int_{t_0}^{t'} |S'(t,k)| dt \qquad (2)$$

Using the claim above and rearranging, we get

$$t_0 - t_1 \leq - \int_{t^*}^{t'} |S(t^*,k)| dt$$

Now suppose $|S(t^*,k)| = 1$, that is, in the original run only one guy points to item k. This must be agent j. This implies $|S(t,k)| = 0$ for $t < t_0$, the time at which j points to k. In particular, we get $C = 0$, and thus $x_{jk} = 1$. We know that $x'_{jk'} > 0$ since j points to $k' \neq k$ in the new run. Therefore, $x'_{jk} < 1$ since $\sum_{k \in I} x_{jk} = 1$. Thus, we may assume $|S(t^*,k)| > 1$, which implies that $t_0 - t_1 < -(t' - t^*)$. Thus,

$$x'_{jk} = t' - t_1 < t^* - t_0 = x_{jk}$$

Proof of Claim B.2. (Sketch) In fact, we claim that for every item $i \neq k$, the subset $S(t,i) \subseteq S'(t,i)$ for $t_0 \leq t < t'$. This can be proved by induction. Suppose the claim is true at some time; it is true at time t_0. The next interesting time t is when some item is i is completely allocated in one of the runs. By our assumption, this time t occurs in the new run since $S'(t,i) \geq S(t,i)$ for $i \neq k$. At this point the agents pointing to i point to different items increasing their corresponding $S'(t,\cdot)$s. The same occurs in the original run albeit at a later time say t''; however, by monotonicity property $S'(t'',i) \supseteq S'(t,i)$, and therefore $|S'(t'',i)| \geq |S(t'',i)|$. For the item k, note that the above argument implies $|S'(t,i) \setminus k| \geq |S(t,i) \setminus k|$, and then after t_1, j enters $S'(t,k)$ as well. $\square$