Randomized Matching Mechanisms Under Single-Minded Preferences

Submission 47

No Institute Given

Abstract. We study the problem of assigning a set of indivisible objects to single-minded agents in the absence of transferable utilities. Under strict ordinal preferences, Random Serial Dictatorship is known to guarantee a desirable set of properties such as Pareto efficiency, strategyproofness, and non-bossiness. However, when agents are able to specify indifferences in their preferences, mechanisms such as RSD are not guaranteed to satisfy these set of properties. We consider two specialized subdomains of preferences, in which agents are respectively single-minded (i.e. only desire one item), and have indifferent preferences of equal size (i.e. they all desire k items). We develop computationally efficient randomized algorithms for use in these two preference domains, that preserve Pareto efficiency, strategyproofness, and non-bossiness. These algorithms are also straightforward to explain or to convert to a single-shot lottery, in contrast to existing approaches.

1 Introduction

We study the problem of assigning a set of objects to a set of agents without the use of money, under the constraint that agents can only receive one object. We consider settings in which agents are single-minded and only care about their top choice as well as settings where agents are indifferent between a subset of objects. These settings capture a variety of resource allocation problems and have been extensively studied in the computer science and economics literature [1, 2, 5–8, 10, 11, 14].

In the *standard assignment* problem (also known as the house allocation problem), each agent is entitled to receive exactly one object from the market. Svensson formulated this setting by assuming that agents have *strict* preferences over objects, and showed that Serial Dictatorship mechanisms are the only social choice rules that satisfy strategyproofness, non-bossiness, and neutrality [15, 16]. Later, Pápai extended these mechanisms to settings where there are potentially more objects than agents (each agent receiving at most one object) with a hierarchy of endowments, generalizing Gale's top trading cycle procedure. This result showed that the hierarchical exchange rules characterize the set of all Pareto efficient, group-strategyproof, and reallocation-proof mechanisms [9].

The proposed mechanisms for strict preferences do not always satisfy the same economic properties when agents' preferences include indifferences between some objects, a domain that arises naturally when some agents consider objects identical but others do not. For example, an agent with a shellfish allergy might consider all seafood restaurants to be equally bad, while other agents might have preferences over them based on the quality of their fare. Bogomolnaia and Moulin first focused on assignment problems under dichotomous preferences where agents classify all objects as acceptable (top choices) or not acceptable [6]. Based upon Bogomolnaia and Moulin's paper, recently Aziz presented a polynomial-time and strategyproof algorithm for finding *deterministic* assignments for house allocation problems with existing tenants under dichotomous preferences [2]. While this approach is efficient, its deterministic nature means that agents are not treated equally ex ante, an important property for ensuring the participation of agents in the mechanism.

Bogomolnaia et al. later studied strategyproof assignments on the full preference domain (i.e. permitting arbitrary combinations of indifference and order among the set of items) and produced a mechanism called the Bi-Polar Serial Dictatorial Rule that satisfies strategyproofness, efficiency, and weak non-bossiness [5]. However, they took an axiomatic approach without proposing an explicit algorithm for finding all these solutions. The proposed mechanism operates by finding a social welfare maximizing assignment under the constraint that agents appearing higher in a priority ordering have the opportunity to seize any item assigned to a lower priority agent. However, the computational complexity of finding such an ordering is not discussed. Cechlárová et al. took an algorithmic approach and presented a polynomial-time algorithm for finding such assignments when agents are allowed to be indifferent between some objects [8]. They further suggested that by randomizing over all priority orderings one can ensure some level of fairness among the agents. However, this clearly requires factorial time to compute the allocation probabilities, because each.

In this paper, we focus on special classes of preferences and develop easy-to-implement and highly efficient algorithms under preferences that include indifference. We first focus on single-minded preferences; a special class of preferences where agents care about their top choice and are indifferent about the rest of the objects. Then, we extend our algorithms to a broader class of preferences with indifferences. Our algorithms are intuitive and use iterative lotteries to assign objects to agents. By restricting the class of preferences, we are able to develop polynomial-time randomized algorithms for the assignment problem under single-minded preferences as well as random assignment with indifferences under some mild assumptions.

Our work also raises interesting questions about the upper and lower bounds of computational cost when allocating items to agents in this domain. We show that in special cases, fair, efficient, and strategyproof assignments can be computed using time polynomial in the number of agents. In contrast, existing results [8] support two-sided matchings as well as allocation, and apply to a more general class of preferences, but require cubic time to compute for a fixed priority ordering or compromise on ex-ante fairness.

2 Model Description & Properties

Let $N = \{1, ..., n\}$ be a set of agents, and $M = \{1, ..., m\}$ be the set of alternatives to be assigned to the agents, where $n = m$. Each agent has a preference ordering over the set of items at time t denoted by $\succ_i$, where $a \succ_i b$ means that agent i strictly prefers item a to item b and $a \sim b$ means that the agent is indifferent between a and b. Let $\mathcal{P}(M)$ or $\mathcal{P}$ denote the class of all linear preferences over M where $|\mathcal{P}| = m!$. We let $\succ = (\succ_1, \ldots, \succ_n) \in \mathcal{P}^n$ denote the preference orderings of the agents, and refer to $\succ$ as the *preference profile*. We write $\succ_{-i}$ to denote $(\succ_1, \ldots, \succ_{i-1}, \succ_{i+1}, \ldots, \succ_n)$, and thus $\succ = (\succ_i, \succ_{-i})$. In addition, we assume that each agent is endowed with a private utility function, u_i, that is consistent with its preference ordering, that is, $u_i(\succ_i, a) > u_i(\succ_i, b)$ if and only if $a \succ_i b$.

A matching $\mu : N \rightarrow M$ is a mapping that assigns a unique item to each agent. A matching is *feasible* if and only if for all $i, j \in N$, $\mu(i) \neq \mu(j)$ when $i \neq j$. We let $\mathcal{M}$ denote the set of feasible matchings. We make the additional assumption that there are no externalities, i.e., an agent's preference for an item depends only on its own idiosyncratic preference and not on the items allocated to other agents.

A *matching mechanism* $\pi : \mathcal{P}^n \rightarrow \Delta(\mathcal{M})$ returns a probability distribution over the set of all possible matchings for each time t. Thus, $\pi(\succ)$ describes the probabilistic assignment of items to agents under preference profile $\succ$. We also write $\pi(\mu|\succ)$ to denote the probability of a particular deterministic matching μ being selected by the mechanism π given the preference profile $\succ^t$. Note that a matching mechanism is based on ordinal preferences (as opposed to cardinal utilities) and returns a (randomized) matching decision given a profile of reported preferences. Since we assume that agents have utility functions, given a preference profile $\succ$ the expected utility for agent i under matching mechanism π is

$$\mathbb{E}_\pi[u_i|\succ] = \sum_{\mu \in \mathcal{M}} u_i(\succ_i, \mu(i))\ \pi(\mu|\succ) \tag{1}$$

An assignment is Pareto efficient if there exists no other allocation that makes at least one agent better without making any of the agents worse off.

Definition 1 (Pareto Efficiency). *An assignment μ prescribed by mechanism π is (ex post) Pareto efficient if there exists no two agents i and j such that $u_i(\mu(j)) > u_i(\mu(i))$ and $u_j(\mu(i)) \geq u_j(\mu(j))$.*

A matching mechanism is said to be strategyproof if no agent can benefit from misreporting its preference. More formally,

Definition 2 (Staretgyproofness). *A matching mechanism π is strategyproof if for any agent i with misreport $\hat{\succ}_i$, while its true preference is $\succ_i$, we have $\mathbb{E}_\pi[u_i|\succ] \geq \mathbb{E}_\pi[u_i|(\hat{\succ}_i, \succ_{-i})]$.*

An important axiomatic property in designing truthful matching mechanisms is *non-bossiness* due to Satterthwaite and Sonnenschein [13]. A mechanism is non-bossy if an agent cannot change the probability of the mechanism allocating an item for some other agent without also changing its own probability distribution over the items.[1] When a mechanism allows agents to be indifferent among different assignments, then non-bossiness is an overly strong condition. For example, a single-minded agent that does not receive its top choice will be indifferent over all items it might be assigned, and so a bossy peer that affects the allocation has no effect on the underlying utilities. Consequently, we require a weaker notion of non-bossiness for agents with single-minded preferences.

Definition 3 (Weak Non-Bossiness (WNB)). *A mechanism is weakly non-bossy if for all* $\succ \in \mathcal{P}^n$ *and agent* $i \in N$*, for all* $\hat{\succ}_i$ *such that* $\mathbb{E}_\pi[u_i | \succ] = \mathbb{E}_\pi[u_i | (\hat{\succ}_i, \succ_{-i})]$*, for all agents* $j \in N$ *we have* $\mathbb{E}_\pi[u_j | \succ] = \mathbb{E}_\pi[u_j | (\hat{\succ}_i, \succ_{-i})]$.

A mechanism that is strategyproof, weakly non-bossy and Pareto efficient is certain to produce good outcomes on the domain of single-minded preferences. Strategyproofness and non-bossiness prevent any strategic behavior by agents while Pareto efficiency ensures that the outcome of a mechanism is stable and desired by all agents.

A Serial Dictatorship mechanism [15, 16] achieves these goals by picking an arbitrary ordering over the agents, and allowing the agents to each select an item in turn. Since a deterministically selected ordering is unfair to the agents at the end, a natural extension is to select the ordering uniformly at random after agents have committed to using the mechanism. The resulting mechanism is called Random Serial Dictatorship (RSD), and maintains the desirable properties of the original deterministic algorithm, while ensuring that agents are treated equally ex ante.

3 Single-Minded Preferences

We first restrict ourselves to players with simple preferences called "single-minded" agents. Single-minded agents have been extensively studied in resource allocation markets such as combinatorial auction design [4].[2] We show that even though the Random Serial Dictatorship mechanism is strategyproof and Pareto efficient for general strict ordinal preferences, it does not always guarantee Pareto efficiency when agents are single-minded. We propose a randomized mechanism that satisfies a set of desirable properties in this setting.

A single-minded agent has a most preferred object (its top choice), that it prefers to all other objects. The agent is is indifferent between all objects that are not its top choice. We assume a single-minded agent receives a positive utility when assigned its top choice and zero utility otherwise.

Definition 4 (Single Minded Preferences). *Agent* i *is single minded if there exists an item* $o_k \in M$ *such that* $\succ_i = o_k \succ o_1 \sim \ldots \sim o_{k-1} \sim o_{k+1} \sim \ldots \sim o_m$.

[1] Formally, a mechanism is non-bossy if for all $\succ \in \mathcal{P}^n$ and agent $i \in N$, for all $\hat{\succ}_i$ such that $\pi_i(\succ) = \pi_i((\hat{\succ}_i, \succ_{-i}))$ we have $\pi(\succ) = \pi((\hat{\succ}_i, \succ_{-i}))$.

[2] Single-minded preferences are special class of dichotomous preferences [6].

While additional criteria like the maximization of social welfare might also be desirable, we will show that natural extensions of existing matching mechanisms like RSD into the single-minded domain do not preserve even the minimal set of axiomatic properties described above, motivating the development of domain-specific mechanisms.

When agents are single-minded, even though the RSD mechanism guarantees strategyproofness, it may prescribe Pareto dominated matchings due to underlying indifferences within agents' preference orderings. Moreover, under single-mindedness RSD no longer satisfies non-bossiness nor weak non-bossiness.

Theorem 1. *Under single-minded preferences, RSD is strategyproof but fails to guarantee Pareto efficiency and weak non-bossiness.*

Proof. We need only to show that there exists a serial dictatorship in the support of RSD that fails to satisfy Pareto efficiency and WNB.

Consider three agents with preferences as following $\succ_1 = a \succ b \sim c, \succ_2 = a \succ b \sim c$, and $\succ_3 = b \succ a \sim c$. Given the following priority ordering $f = (1, 2, 3)$, the serial dictatorship runs as follows: agent 1 chooses a, agent 2 is indifferent between b and c so it may choose b and agent 3 gets the remaining object c. Since agent 2 is indifferent between b and c (meaning that $u_2(\succ_2, b) = u_2(\succ_2, c)$, for all utility models), but agent 3 prefers object b to c, *i.e.* $u_3(\succ_3, b) > u_3(\succ_3, c)$. Thus, a matching that assigns c to agent 2 and b to agent 3 strictly improves agent 3's utility without making agent 1 worse off, implying that even though RSD is strategyproof, the induced matching is not Pareto efficient.

For WNB, in the above serial dictatorship, agent 2's decision to pick object b (as opposed to c) does not change its utility but strictly decreases agent 3's utility. RSD is a uniform randomization over all serial dictatorship mechanisms, implying that RSD is neither Pareto efficient nor WNB.

While other mechanisms exist for this domain, they are more complicated, and more computationally intensive than RSD [6,8]. Since this preference domain is quite simple, it is reasonable to suppose that there exist mechanisms that both preserve the desired axiomatic properties and compute outcomes quickly and simply.

Inspired by the RSD mechanism, we propose a mechanism, called Random Equivalence Class Assignment (RECA) that ensures our desirable properties. Before describing the mechanism, we define equivalence classes for single-minded agents, where agents with similar top choices are assigned to the same equivalence classes.

Definition 5. *Given a set of agents N, an equivalence class for object $x \in M$ is defined as follows*

$$C_x = \{i \in N : top(\succ_i) = x\} \tag{2}$$

Given the definition of equivalence classes, Algorithm 1 shows the steps of our RECA mechanism. The RECA mechanism (Algorithm 1) allocates objects to agents with single-minded preferences as follows:

Algorithm 1: Random Equivalence Class Assignment (RECA)

```
Input: A preference profile ≻ of single-minded agents
Output: A matching μ according to reported preferences ≻
// Setting up equivalence classes
A ← ∅ ;                          // Set of assigned agents, initially empty.
B ← ∅ ;                          // Set of assigned objects, initially empty.
foreach (x ∈ M) do
    C_x ← {} ;
foreach (i ∈ N) do
    C_top(≻_i) ← C_top(≻_i) ∪ {i} ;
foreach ( Equivalence class C_x) do
    i ← randomly choose an agent from C_x ;
    μ(i) = x ;              // Assign object x to the randomly chosen agent.
    A ← A ∪ {i} ;                   // Add agent to the set of assigned agents.
    B ← B ∪ {x} ;                 // Add object to the set of assigned objects.
foreach ( object x ∈ M \ B) do
    i ← randomly choose an agent from N \ A ;
    μ(i) = x ;
    A ← A ∪ {i} ;                   // Add agent to the set of assigned agents.
    B ← B ∪ {x} ;                 // Add object to the set of assigned objects.
return μ
```

- Initialization: the mechanism creates equivalent classes based on the objects.
- Each agent is assigned to an equivalence class that corresponds to its top choice item.
- For each equivalence class, the mechanism assigns the object associated to that equivalence class to a randomly chosen agent from the set of agents in that equivalence class.
- While there are unassigned objects, the mechanism randomly assigns an object from the set of remaining objects to an agent from the set of remaining agents.

We let π be the random matching prescribed by RECA where $\pi(\succ)$ denotes the fractional probabilities before the realization of the matching decision and $\pi_{i,j}(\succ)$ denotes the probability that agent i receives object j under RECA.

We show that our algorithm for matching objects to a set of agents with single-minded preferences satisfies strategyproofness, Pareto efficiency, and weak non-bossiness.

Theorem 2. *Random Equivalence Class Assignment (RECA) is strategyproof, Pareto efficient, and weakly non-bossy.*

Proof. **Strategyproofness**: since agents are single-minded, they are indifferent between all objects that are not ranked first. Consider a case where agent i misreports its top object, and thus, $top(\succ_i) \neq top(\hat{\succ}_i)$. If no other agent has the same top object, then agent i is in $C_{top(\hat{\succ}_i)}$ and receives that object with certainty. By Algorithm 1 it's clear that $\mathbb{E}_\pi[u_i|\succ] = \mathbb{E}_\pi[u_i|(\hat{\succ}_i, \succ_{-i})]$. If there exists some other agent j with $top(\succ_j) = top(\hat{\succ}_i)$, but no other agent's top choice is $top(\succ_i)$, *i.e.* no other agent wants agent i's top choice according to its truthful preference, then agent i has some probability of receiving its top choice. However, since no one else wants agent i's top choice then $C_{top(\succ_i)}$ consists of only agent i, which means that agent i would receive $top(\succ_i)$ with certainty under a truthful report. This shows that under no condition a strategic agent could benefit from misreporting.

Pareto efficiency: Let μ denote the assignment returned by RECA, and for simplicity let $u_i(\mu(i)) = u_i(\succ_i, \mu(i))$ denote agent i's utility for $\mu(i)$. Suppose that RECA is not Pareto efficient. This means that there are at least two agents i and j such that $u_i(\mu(j)) > u_i(\mu(i))$ and $u_j(\mu(i)) \geq u_j(\mu(j))$.

The inequality $u_j(\mu(i)) \geq u_j(\mu(j))$ implies that either agent j is indifferent between $\mu(i) \sim \mu(j)$ or $\mu(i) \succ_j \mu(j)$. If $\mu(i) \succ_j \mu(j)$ then changing the assignment of agent j with i would make agent j strictly worse off, which contradicts the assumption. Let's assume that agent j is indifferent between $\mu(i)$ and $\mu(j)$, *i.e.* $u_j(\mu(i)) = u_j(\mu(j))$. Thus, it must be the case that $top(\succ_j) \neq \mu(i) \neq \mu(j)$.

The strict inequality in $u_i(\mu(j)) > u_i(\mu(i))$ implies that $\mu(j) \succ_i \mu(i)$, and thus by single-mindedness of preferences it must be the case that $\mu(j) = top(\succ_i)$. Therefore, we can conclude that agent i belongs to the equivalence class of $C_{\mu(j)}$. Since $\mu(i) \neq \mu(j)$ by definition, it's clear that Algorithm 1 has not randomly selected agent i from $C_{\mu(j)}$, and agent i was assigned an object from the set of unassigned objects. Thus, $\mu(j)$ must be assigned to an agent with k with $top(\succ_k) = \mu(j)$ who strictly prefers $\mu(j)$ to $\mu(i)$. However, we know that $\mu(j)$ is assigned to j and we showed that $u_j(\mu(i)) = u_j(\mu(j))$, which contradicts the assumption.

Weak non-bossiness: The proof follows from the steps in Algorithm 1. A mechanism is weakly bossy, if an agent reports a preference $\hat{\succ}_i$ such that its expected utility remains the same $\mathbb{E}_\pi[u_i|\succ] = \mathbb{E}_\pi[u_i|(\hat{\succ}_i, \succ_{-i})]$ while at least one another agent j's expected utility is $\mathbb{E}_\pi[u_j|\succ] > \mathbb{E}_\pi[u_j|(\hat{\succ}_i, \succ_{-i})]$.

If agent i reports a preference such that $top(\hat{\succ}_i) \neq top(\succ_i)$, then by Algorithm 1 it loses it chance of getting selected from $C_{top(\succ_i)}$, and thus $\mathbb{E}_\pi[u_i|\succ] < \mathbb{E}_\pi[u_i|(\hat{\succ}_i, \succ_{-i})]$. For all other objects, either the object is a top choice of agent j which would be randomly assigned to an agent from $C_{top(\succ_j)}$, or the object is not a top choice of agent j, implying that $\mathbb{E}_\pi[u_j|\succ] = \mathbb{E}_\pi[u_j|(\hat{\succ}_i, \succ_{-i})]$. Thus, no agent can reduce the expected utility of another agent while receiving the same expected utility, which shows that RECA is weakly non-bossy.

Computing RSD probabilities for general preferences is #P-complete [3], because the associated counting problem is intractable and requires $n!$ deterministic matchings to be computed. In the next theorem, we show that although computing RSD probabilities is intractable, computing RECA fractional probabilities can be done in polynomial time.

Theorem 3. *Given single-minded preferences, computing RECA probabilities can be done in polynomial time.*

Proof. It is sufficient to show that the probability that an agent receives each of the objects can be written in a closed form. Let c denote the number of non-empty equivalence classes. Thus, given m objects, according to Algorithm 1 the number of objects that are ranked first by no agent is $m - c'$.

Let $\pi_{i,j} = \pi_{i,j}(\succ)$ denote the probability that agent i receives object j under RECA. For each agent, the probability of receiving the objects in M according to RECA is:

- In each equivalence class, the mechanism assigns the associated object to a randomly chosen agent from the set of agents in that class. Thus, $\pi_{i,top(\succ_i)} = \frac{1}{|C_{top(\succ_i)}|}$.
- All objects that are ranked first by at least one agent get assigned to a member of their equivalence classes. Therefore, for $m - c'$ remaining objects, the probability that agent i receives object j', is $\pi_{i,j'} = \frac{(1-\pi_{i,top(\succ_i)})}{m-c'}$.
- Lastly, no agent can have a chance to receive an object that is not in its own equivalence class, but it is ranked first by another agent. For each object j which is ranked first by at least one agent, assign probability zero to all agents that $top(\succ_i) \neq j$, *i.e.*, for each j where $\exists i' \in N$ such that $j = top(\succ_{i'})$, $\pi_{i,j} = 0$ for all i such that $j \neq top(\succ_i)$.

Each of the above steps can be done in at most $O(n^2)$. Thus, RECA probabilities for single-minded agents can be computed in polynomial time, which completes the proof.

4 Indifference Classes

In this section we extend our allocation mechanism to a more generic class of preferences, namely preferences with indifferences.

Achieving strategyproofness in the presence of ties is quite challenging. Cechlárová et al. [8] developed truthful algorithms for the house allocation problem when there are ties. Their algorithms essentially are extensions of Serial Dictatorship mechanisms. Given a fixed priority ordering, Cechlárová et al. proposed an algorithm that runs in $O(n^3)$ [8]. It is possible to realize a randomized version of this algorithm efficiently, but agents may have difficulty computing the overall probabilities of reveiving different items, because doing so requires enumerating the probability of winning an item under all $n!$ priority orderings, and thus is #P-complete [3, 12]. This may discourage agents from participating in the randomized version of such a mechanism.

In contrast, we seek to develop intuitive algorithms that satisfy axiomatic properties, treat agents equally ex ante, are easy to explain and also require polynomial time to reason over.

4.1 Random Indifference Class Assignment

Let $\succ_i = (c_i^1 \succ c_i^2 \succ \ldots)$ denote agent i's preference ordering with indifference classes such that c_i^k is the kth ranked indifference class for agent i. A key assumption in our model is that agents have equal size indifference classes, i.e. for any ranking k, for all $i, j \in N$ we have $|c_i^k| = |c_j^k|$. Consequently, we need to update the definition of an equivalence classes.

An equivalence class for object $x \in M$, given the kth ranked indifference class is

$$C_x^k = \{i \in N : x \in c_i^k\} \tag{3}$$

Given the success of RECA in the domain of single-minded preferences, it is natural to wonder whether RECA can also be applied fruitfully to the problem of allocation with indifference classes of equal size. The natural extension, which we term RECA-e, would construct the equivalence classes C_x^k, and then sample a permutation over the classes, selected uniformly at random. The classes would be resolved in the order specified by the permutation. Resolution consists of assigning the object k associated with the class to one of the agents i in the class (selected uniformly at random), and then removing i from *every* remaining unresolved equivalence class. RECA-e is still strategyproof, as can be demonstrated by considering the (lack of) benefits to be gained from misreporting a false preferences in any of the possible permutations. However, it is straightforward to show that RECE-e is no longer Pareto efficient, rendering it unsuitable for the purposes of this paper. Consider the following counter example: Agent i's top set contains items $\{1, 2\}$, agent j's top set contains items $\{1, 2\}$, and agent k's top set contains the items $\{1, 3\}$. If item 1 is resolved first, there is a $\frac{1}{3}$ chance that the item is assigned to agent k, after which agents i and j must split item 2 between them. In contrast, if item 3 is resolved first, it is automatically assigned to agent k, and items 1 and 2 can be resolved in any order to achieve a Pareto efficient outcome. While this example shows that RECA-e is not Pareto efficient, it also suggests a natural refinement that *does* satisfy all properties of interest to us.

The mechanism in Algorithm 2 (RICA) is an extension of RECA that works by sorting equivalence classes in the increasing order for assigning objects that belong to the same indifference classes. It works by iteratively running lotteries for objects with lower demand (smaller equivalence classes), removing the assigned agents from equivalence classes, and then choosing the next smallest equivalence class (breaking ties at random). After all possible assignments are made for an indifference class then the mechanism moves to the next ranked indifference class.

Theorem 4. *RICA is strategyproof, Pareto efficient, and non-bossy.*

Proof. The proofs for non-bossiness and strategyproofness follow from arguments similar to those used in the case with single-minded preferences in Theorem 2.

For strategyproofness, in each iteration for the assignment of objects in the kth indifference classes, an agent can either participate in the corresponding lottery for C_x^k such that $x \in c_i^k$ or participate in another lottery for an object x' such that $x' \prec_i x$. By definition, we have that for all $i, j \in N$ we have $|c_i^k| = |c_j^k|$. Thus, an agent cannot benefit from reporting a less preferred object in round k.

Algorithm 2: Random Indifference Class Assignment (RICA)

Input: A preference profile $\succ$ of single-minded agents
Output: A matching μ according to reported preferences $\succ$
`// Setting up equivalence classes`
$A \leftarrow \emptyset$; `// Set of assigned agents, initially empty.`
$B \leftarrow \emptyset$; `// Set of assigned objects, initially empty.`
foreach $k : 1 \ldots \ell$ **do**
  foreach $(i \in N)$ **do**
    foreach $x \in c_i^k$ **do**
      $C_x^k \leftarrow C_x^k \cup \{i\}$; `// Initializing all indifference classes.`
foreach $k : 1 \ldots \ell$ **do**
  `//` ℓ `is the least preferred indifference class.`
  Sort all C_x^k according to size in ascending order ;
  foreach (*Equivalence class* C_x^k *in ascending order*) **do**
    $i \leftarrow$ randomly choose an agent from C_x^k ;
    $\mu(i) = x$; `// Assign object` x `to the randomly chosen agent.`
    $A \leftarrow A \cup \{i\}$; `// Add agent to the set of assigned agents.`
    $B \leftarrow B \cup \{x\}$; `// Add object to the set of assigned objects.`
    $C_x^{k'}$ from the set of indifference classes;
    Remove $\{i\}$ from all $C_y^{k'}$, where $y \in M$;
    Update sorted ordering of C_x^k ;
foreach (*object* $x \in M \setminus B$) **do**
  $i \leftarrow$ randomly choose an agent from $N \setminus A$;
  $\mu(i) = x$;
  $A \leftarrow A \cup \{i\}$; `// Add agent to the set of assigned agents.`
  $B \leftarrow B \cup \{x\}$; `// Add object to the set of assigned objects.`
return μ

For Pareto efficiency, it's crucial for the mechanism to start the randomized assignment from equivalence classes with smaller size in each iteration. That is, starting from $k = 1$ (top choice indifference class) upwards, RICA first start from the equivalence class with the smallest nonempty number of agents. This property guarantees that the objects in less demand get assigned first. For contradiction assume that there are two agents i, j such that $u_i(\mu(j)) > u_i(\mu(i))$ and $u_j(\mu(i)) \geq u_j(\mu(j))$. This means that during the allocation of kth ranked indifference class, agent j is assigned an object prior to agent i and agent i was assigned an object during $(k+1)$th round. However, if agent i liked object $\mu(j)$ then it should also have participated in the assignment of object $\mu(j)$ in round k unless agent i was assigned an object o in round k such that $o \succ_i \mu(j)$, which is a contradiction.

Note that RICA guarantees strategyproofness (in its strongest sense) when agents' indifference classes are the same size, that is, when $|c_i^k| = |c_j^k|$ for all $i, j \in N$. However, if we allow for general preferences with indifferences then RICA becomes susceptible to manipulation even when only two indifference classes are permitted. Take for example the following preferences: $\succ_1$: $a \succ b, c$, $\succ_2$: $b, c \succ a$, and $\succ_3$: $a, b, c \succ \emptyset$. Agent 1 can misreport its preference as $\hat{\succ}_1 : a, b, c \succ \emptyset$ and inflate the other equivalence classes to increase the probability of being assigned its top choice. In this example, agent 1's probability of being assigned object a increases from $\frac{1}{6}$ to $\frac{1}{2}$.

Theorem 5. *Both implementing RICA and computing the probability that a particular agent will receive a particular item under RICA can be done in polynomial time.*

Proof. In Algorithm 2, the initialization of equivalence classes takes $O(n \cdot m)$. For each k, the algorithm sorts the indifference classes in $O(m \log m)$ and this is repeated for ℓ indifference classes. In the worst case, we have $\ell = m$ indifference classes, and thus, the overall running time is $O(n \cdot m + m^2 \log m)$. The rest of the argument follows from the proof of Theorem 3.

5 Conclusion

This paper introduced novel and efficient algorithms for the assignment of items to agents under two specialized preference models involving indifference. While past approaches to the full preference domain have provided solutions that are strategyproof, Pareto efficient and non-bossy, these approaches have either not considered the question of computational costs [6], or are comparatively slow to find deterministic allocations and extremely slow to find the probabilistic allocations needed to treat agents equally and to provide a simple, one shot mechanism [8].

Our mechanisms for single minded preferences (RECA) and indifferent preferences of equal length (RICA) are strategyproof, Pareto efficient and non-bossy. However, they are also efficient to compute, even when operating with randomization to treat agents equally. The probabilities needed to convert them to one shot mechanisms are also straightforward and computationally inexpensive to compute.

The proposed algorithms may have specialized applicability to particular real world problem domains. However, more interestingly, they illustrate that it is possible to find allocations quite simply and inexpensively for at least some of the problems in the full preference domain. A natural question for future consideration is whether there are other classes for which this problem is just as easy. The characterization of such classes may provide insights into the difficulty of working with the full domain of preferences in general.

References

1. Atila Abdulkadiroğlu, Parag A. Pathak, and Alvin E. Roth. Strategy-proofness versus efficiency in matching with indifferences: redesigning the new york city high school match. Technical report, National Bureau of Economic Research, 2009.

2. Haris Aziz. Mechanisms for house allocation with existing tenants under dichotomous preferences. *CoRR*, abs/1603.03073, 2016.
3. Haris Aziz, Felix Brandt, and Markus Brill. The computational complexity of random serial dictatorship. *Economics Letters*, 121(3):341–345, 2013.
4. Liad Blumrosen and Noam Nisan. Combinatorial auctions. *Algorithmic Game Theory*, 267:300, 2007.
5. Anna Bogomolnaia, Rajat Deb, and Lars Ehlers. Strategy-proof assignment on the full preference domain. *Journal of Economic Theory*, 123(2):161–186, 2005.
6. Anna Bogomolnaia and Hervé Moulin. Random matching under dichotomous preferences. *Econometrica*, pages 257–279, 2004.
7. Eric Budish, Yeon-Koo Che, Fuhito Kojima, and Paul Milgrom. Designing random allocation mechanisms: Theory and applications. *The American Economic Review*, 103(2):585–623, 2013.
8. Katarína Cechlárová, Pavlos Eirinakis, Tamás Fleiner, Dimitrios Magos, David F Manlove, Ioannis Mourtos, Eva Ocel'áková, and Baharak Rastegari. Pareto optimal matchings in many-to-many markets with ties. In *International Symposium on Algorithmic Game Theory*, pages 27–39. Springer, 2015.
9. Szilvia Pápai. Strategyproof assignment by hierarchical exchange. *Econometrica*, 68(6):1403–1433, 2000.
10. Alvin E. Roth and John H. Kagel. *The handbook of experimental economics*, volume 1. Princeton university press Princeton, 1995.
11. Alvin E. Roth, Tayfun Sönmez, and M. Utku Ünver. Kidney exchange. *The Quarterly Journal of Economics*, 119(2):457–488, 2004.
12. Daniela Saban and Jay Sethuraman. The complexity of computing the random priority allocation matrix. *Mathematics of Operations Research*, 40(4):1005–1014, 2015.
13. Mark A. Satterthwaite and Hugo Sonnenschein. Strategy-proof allocation mechanisms at differentiable points. *The Review of Economic Studies*, pages 587–597, 1981.
14. Tayfun Sönmez and M. Utku Ünver. Course bidding at business schools. *International Economic Review*, 51(1):99–123, 2010.
15. Lars-Gunnar Svensson. Queue allocation of indivisible goods. *Social Choice and Welfare*, 11(4):323–330, 1994.
16. Lars-Gunnar Svensson. Strategy-proof allocation of indivisible goods. *Social Choice and Welfare*, 16(4):557–567, 1999.