Towards a Characterization of Random Serial Dictatorship

Felix Brandt Matthias Greger René Romen

Abstract

Random serial dictatorship (RSD) is a randomized assignment rule that—given a set of n agents with strict preferences over n houses—satisfies equal treatment of equals, *ex post* efficiency, and strategyproofness. For $n \leq 3$, Bogomolnaia and Moulin [3] have shown that RSD is characterized by these axioms. Extending this characterization to arbitrary n is a long-standing open problem. By weakening *ex post* efficiency and strategyproofness, we reduce the question of whether RSD is characterized by these axioms for fixed n to determining whether a matrix has rank $n^2 n!^n$. We leverage this insight to prove the characterization for $n \leq 5$ with the help of a computer.

1 Introduction

Assigning objects to individual agents is a fundamental problem that has received considerable attention by computer scientists as well as economists [e.g., 7, 20, 11, 4]. The problem is known as the *assignment problem*, the *house allocation problem*, or *two-sided matching with one-sided preferences*. In its simplest form, there are n agents, n houses, and each house needs to be allocated to exactly one agent based on the strict preferences of each agent over the houses. Applications are diverse and include assigning dormitories to students, jobs to applicants, processor time slots to jobs, parking spaces to employees, offices to workers, etc.

A class of simple, well understood, and often applied deterministic assignment rules are *serial dictatorships*, which are based on a fixed priority order over the agents that is independent of the reported preferences. The agent with the highest priority gets to pick her most preferred house, then the second agent chooses her most preferred among the remaining houses, and so on. Serial dictatorships are guaranteed to return a Pareto efficient allocation. On top of that, they are neutral (when houses are permuted, the assignment is permuted accordingly), nonbossy (an agent cannot affect the assignment to other agents without changing the house allocated to herself), and strategyproof (no agent can misreport her preferences in order to obtain a more preferred house). Unsurprisingly, like any deterministic rule, serial dictatorships are highly unfair. For example, consider two agents who both prefer house h_1 to h_2. Any deterministic rule strongly discriminates the agent who receives h_2.

Fairness is typically established by allowing for *probabilistic* assignment rules where each agent receives each house with some probability and the probabilities sum up to 1 for each agent and each house. The resulting probability matrix is called a bistochastic matrix. The Birkhoff-von Neumann theorem shows that every bistochastic matrix can be decomposed into a convex combination of permutations matrices. As a consequence, every probabilistic assignment rule can be implemented in practice by picking a deterministic assignment rule at random. The two most prominent probabilistic assignment rules are *random serial dictatorship* (RSD)—also known as *random priority*—and the *probabilistic serial* rule [3].

A natural way to obtain a randomized assignment rule is to apply a deterministic rule to every permutation of the agents' roles and then uniformly randomize over all of these $n!$ deterministic assignments. Such a *symmetrization* ensures that "equals are treated equally". In fact, RSD is defined as the symmetrization of all serial dictatorships and has been shown to be equivalent to the symmetrization of Gale's top trading cycles mechanism [1, 9]. Svensson [21] showed that any deterministic, strategyproof, nonbossy, and neutral assignment rule is serially dictatorial, implying that the symmetrization of any such rule has to coincide with *RSD*. Pápai [13] and Pycia and Ünver [17] have characterized broader classes of deterministic assignment rules by replacing neutrality with efficiency.

The main axiomatic advantage of *RSD* is that it satisfies strategyproofness while also guaranteeing efficiency and fairness to some extent. While *RSD* does satisfy *ex post* efficiency, it violates a stronger efficiency notion called ordinal efficiency or *SD*-efficiency [3]. In fact, Bogomolnaia and Moulin showed that strategyproofness and equal treatment of equals are incompatible with ordinal efficiency. Furthermore, they observed that *RSD* only satisfies a weak notion of envy-freeness. The probabilistic serial rule, on the other hand, satisfies ordinal efficiency and envy-freeness but violates strategyproofness.

A characterization of *RSD* via equal treatment of equals, *ex post* efficiency, and strategyproofness is a long-standing open problem [see, e.g., 14, 15] and would cement its pivotal role in settings where strategyproofness is indispensable.

Unfortunately, to the best of our knowledge, there does not even exist a characterization of all *deterministic*, strategyproof, and efficient assignment rules [cf. 21]. Furthermore, Aziz et al. [2] and Saban and Sethuraman [18] showed that it is NP-complete to decide whether an agent receives a given house with positive probability under *RSD*, stressing its combinatorial intricacy.

Pycia and Troyan [16] recently showed that *RSD* is characterized by symmetry, efficiency, and obvious strategyproofness among all assignment rules that, roughly speaking, can be represented as a symmetrization of an extensive-form game where in each stage, one agent is allowed to pick one house from a subset of the remaining houses or "pass" on this opportunity. Furthermore, Pycia and Troyan [15] point out that equal treatment of equals, *ex post* efficiency, and strategyproofness do not suffice to characterize *RSD* when using a stronger equivalence notion that interprets two rules as different if they produce different distributions over deterministic assignments, even when the probabilistic assignment is still the same. By contrast, we consider two rules as equivalent if, for each profile, they return the same probabilistic assignment.

In this paper, we use a linear algebraic approach to show that the desired characterization holds for $n \leq 5$. After introducing the necessary notation and central axioms in Section 2, we reduce the question of checking whether the characterization holds to determining the rank of a matrix by weakening *ex post* efficiency and strategyproofness in Section 3. Based on this idea, we devise an algorithm that determines the rank of the given matrix and use it to prove the characterization for $n \leq 5$ with the help of computer in Section 4. Finally, our results and further insights are summarized in Section 5.

2 Preliminaries

Let N be a set of agents and H a set of houses with $|N| = |H| = n$. A *preference profile* R associates with each agent $i \in N$ a preference ordering $\succ_i$ over the houses. The set of all preference profiles is denoted by $\mathcal{R}$. *Random assignments* are represented by *bistochastic matrices* $(M_{i,h})_{i\in N, h\in H}$ where $M_{i,h} \geq 0$ and $\sum_{h'\in H} M_{i,h'} = \sum_{i'\in N} M_{i',h} = 1$ for all $i \in N$ and $h \in H$. The *support* of a random assignment M is the set of agent-house pairs (i, h) for which $M_{i,h} > 0$. Whenever $M_{i,h} \in \{0, 1\}$ for all agent-house pairs (i, h), M is a permutation matrix and represents a *deterministic assignment*.

A probabilistic assignment rule f maps each profile R to a bistochastic matrix $f(R)$ where, with slight abuse of notation, the entry $f(R, i, h)$ in the ith row and hth column of the matrix corresponds to the probability of agent i receiving house h in profile R.

In the following, we formally define *RSD* and the axioms required for the characterization.

Definition 1. Given a profile $R \in \mathcal{R}$, a deterministic assignment M is (Pareto) *efficient* if there exists no deterministic assignment $M' \neq M$ such that for all $i \in N$ and $h, h' \in H$ with $h \neq h'$, $M'_{i,h'} = M_{i,h} = 1$ implies $h' \succ_i h$. An assignment rule is *ex post efficient* if for all $R \in \mathcal{R}$, $f(R)$ can be represented as a convex combination of efficient deterministic assignments.

Let Π be the set of all (priority) orders over the agents. Denote serial dictatorship for a specific priority order $\pi \in \Pi$ by SD_π. For a given profile R, each deterministic efficient assignment coincides with the outcome of a serial dictatorship on R [see, e.g., 10]. Therefore, an assignment rule satisfies *ex post* efficiency if for all $R \in \mathcal{R}$, there exist weights $\lambda_\pi^R \geq 0$ with $\sum_{\pi\in\Pi} \lambda_\pi^R = 1$ such that $f(R) = \sum_{\pi\in\Pi} \lambda_\pi^R SD_\pi(R)$.

RSD can now be defined by choosing $\lambda_\pi^R = 1/n!$ for every π and R, i.e.,

$$RSD(R) = \sum_{\pi\in\Pi} \frac{1}{n!} SD_\pi(R).$$

Furthermore, we say that a rule coincides with *RSD* if it returns the same random assignment as *RSD* for each profile.

It turns out that a weak variant of *ex post* efficiency suffices to obtain a characterization for $n \leq 5$. This variant merely requires that for each profile the support of the resulting random assignment coincides with that of some *ex post* efficient random assignment. In other words, the support has to be a subset of that of *RSD*.

Definition 2. An assignment rule f is *support efficient* if for all $R \in \mathcal{R}$, $i \in N$, and $h \in H$, $f(R, i, h) = 0$ whenever $SD_\pi(R, i, h) = 0$ for all $\pi \in \Pi$. Equivalently, f is support efficient if for all $R \in \mathcal{R}$, $i \in N$, and $h \in H$, $RSD(R, i, h) = 0$ implies $f(R, i, h) = 0$.

Support efficiency and *ex post* efficiency are equivalent for $n = 3$. A proof can be found in Appendix A. We now give an example for 4 agents in which support efficiency is strictly weaker than *ex post* efficiency.

Example 1. Let the preference relations of agents 1 and 2 be $h_1 \succ h_2 \succ h_3 \succ h_4$ and $h_2 \succ h_1 \succ h_3 \succ h_4$ be the preferences of agents 3 and 4. Consider the random assignment where agents 1 and 2 receive the lottery $p(h_1) = 0$, $p(h_2) = \frac{1}{2}$, $p(h_3) = p(h_4) = \frac{1}{4}$ and agents 3 and 4 receive the lottery $p(h_1) = \frac{1}{2}$, $p(h_2) = 0$, $p(h_3) = p(h_4) = \frac{1}{4}$. This assignment violates *ex post* efficiency because each efficient deterministic assignment assigns either h_1 to agent 1 or 2 or it assigns h_2 to agent 3 or 4. Since agents 1 and 2 never receive h_1 and agents 3 and 4 never receive h_2 from the random assignment, it cannot be represented as a distribution over efficient deterministic assignments. The assignment satisfies support efficiency since each house can go to each agent in some efficient deterministic assignment.

To judge whether an agent i is able to beneficially misreport her preferences, we, analogously to Bogomolnaia and Moulin [3], assume that agent i has a von Neumann-Morgenstern utility function u_i which is consistent with $\succ_i$. This means that there exist $u_i \colon H \to \mathbb{R}$ such that $u_i(f(R)) = \sum_{h\in H} u_i(h) f(R, i, h)$, and $u_i(h_k) > u_i(h_l)$ if and only if $h_k \succ_i h_l$. Since the concrete utility function is unknown, a manipulation counts as beneficial if there exists a utility function u_i consistent with $\succ_i$ for which it is beneficial. A rule without such manipulation incentives is called strategyproof.[1]

Definition 3. An assignment rule f is *strategyproof* if for all $R, R' \in \mathcal{R}$ with $\succ_j = \succ'_j$ for all $j \in N \setminus \{i\}$, $\sum_{h'\succ_i h} f(R, i, h') \geq \sum_{h'\succ_i h} f(R', i, h')$ for every $h \in H$.

[1]This version of strategyproofness for probabilistic assignment rules is sometimes also called (strong) *SD*-strategyproofness [see, e.g., 5].

To implement strategyproofness, we leverage a result from Gibbard [8], which shows that a mechanism is strategyproof if and only if it is localized and nonperverse. In particular, it suffices to consider swaps of two houses that are adjacent in the manipulator's ranking.[2]

Definition 4. Let $R, R' \in \mathcal{R}$, $i \in N$, and $h_k, h_l \in H$ such that $\succ_j = \succ'_j$ for all $j \in N \setminus \{i\}$ and $\succ'_i = \succ_i \setminus \{(h_k, h_l)\} \cup \{(h_l, h_k)\}$. An assignment rule f is

- *localized* if $f(R, i, h) = f(R', i, h)$ for all $h \in H \setminus \{h_k, h_l\}$, and
- *nonperverse* if $f(R, i, h_k) \geq f(R', i, h_k)$ and $f(R, i, h_l) \leq f(R', i, h_l)$

It turns out that weakening strategyproofness to localizedness is sufficient for the characterization to hold for $n \leq 5$ and eliminates the inequality constraints imposed by nonperverseness.

Definition 5. An assignment rule f satisfies *equal treatment of equals* if for all $R \in \mathcal{R}$ and $i, j \in N$ with $\succ_i = \succ_j$, $f(R, i, h) = f(R, j, h)$ for all $h \in H$.

Thus, equal treatment of equals ensures that agents with the same preferences receive the same assignment.

Finally, we introduce a natural property that is helpful for reducing the number of profiles a rule needs to be defined on.

Definition 6. An assignment rule f is *symmetric* if for all $R \in \mathcal{R}$, any permutation of the agents $\pi : N \to N$ we have $\pi \circ f(R) = f(\pi \circ R)$ and for any permutation of the houses $\tau : H \to H$ we have $\tau \circ f(R) = f(\tau \circ R)$. Here, π permutates the rows and τ permutates the columns of R and $f(R)$.

Loosely speaking, a symmetric rule does not take into account the identities of agents and houses.

Remark 1. The two conditions of symmetry are known as anonymity and neutrality in the more general domain of social choice [see, e.g., 22]. Within the assignment domain, anonymity cannot be considered in isolation because agents are indifferent between assignments in which they receive the same house. Viewing agents as voters and deterministic assignments as alternatives, permutations via neutrality allow for permuting assignments, not houses. Permuting two voters i and j via anonymity results in an "illegal" assignment profile because agent i is indifferent between assignments in which agent j receives the same house and vice versa. This can be rectified by permuting assignments accordingly. As a consequence, anonymity should only be considered in conjunction with neutrality in the assignment domain.

Note that symmetry is a stronger axiom than equal treatment of equals.

Proposition 1. *Every symmetric assignment rule satisfies equal treatment of equals.*

The proof is deferred to the appendix.

To see that equal treatment of equals does not imply symmetry, consider $n = 2$ and the assignment rule f with $f(R) = RSD(R)$ for the two profiles where both agents have the same preferences. For the other two profiles R' and R'' where both agents have different preferences let $f(R', 1) = (1, 0)$ and $f(R'', 1) = (1, 0)$. Clearly, f satisfies equal treatment of equals. However, moving from R' to R'' by permuting the two houses does not permute the assignments. In both profiles, agent 1 receives h_1, contradicting $\tau \circ f(R') = f(\tau \circ R') = f(R'')$.

[2] Gibbard considers the general social choice domain. Mennle and Seuken [12] have rediscovered this equivalence in the context of random assignment.

Symmetry imposes an equivalence class structure on $\mathcal{R}$ that allows f to be well-defined by only defining it on the set of canonical profiles $\mathcal{R}^* \subset \mathcal{R}$ which contains one representative profile for each equivalence class that is chosen according to some predefined order over $\mathcal{R}$. We will show that positive results for $\mathcal{R}^*$ carry over to $\mathcal{R}$ without imposing symmetry, a necessary simplification step given that $|\mathcal{R}| = n!^n$.

3 A linear algebraic view on the problem

Our overall goal in this section is to describe the set of all rules that satisfy equal treatment of equals, *ex post* efficiency, and strategyproofness by a system of linear equations.

To this end, note that, for fixed n, all axioms except *ex post* efficiency are defined and can be represented by constraints in terms of a vector $\mathbf{x} = \left(\mathbf{x}_{(R,i,h)}\right) \in \mathbb{R}^{n^2 n!^n}$ where $\mathbf{x}_{(R,i,h)}$ corresponds to $f(R,i,h)$. By contrast, efficiency constraints require us to represent an assignment rule f by a vector $\mathbf{x} = \left(\mathbf{x}_{(R,\pi)}\right) \in \mathbb{R}^{n!n!^n}$ where $\mathbf{x}_{(R,\pi)}$ corresponds to the weight λ_π^R of SD_π in profile R.

Generally, it is possible to also represent the other axioms in terms of $\mathbf{x}_{(R,\pi)}$, e.g., for equal treatment of equals, one has to find the set of all combinations of serial dictatorships that yield the same probabilistic assignment for both agents for each profile where two agents i and j have the same preferences. This can be achieved by requiring that the sum of the weights of all serial dictatorships where i receives house h has to equal the sum of the weights of all serial dictatorships where j gets h. However, the representation of f in terms of $\left(\mathbf{x}_{(R,\pi)}\right)$ is not unique [see, e.g., 15] and requires $n!n!^n$ instead of $n^2 n!^n$ variables.

Weakening *ex post* efficiency to support efficiency enables the representation of efficiency via $f(R,i,h)$. On top of that, we also weaken strategyproofness to localizedness due to the fact that nonperverseness is the only axiom (apart from the non-negativity part of the bistochastic matrix constraints) that cannot be written in terms of linear equations.

Conjecture 1. *RSD is the only assignment rule that satisfies equal treatment of equals, support efficiency, and localizedness.*

Proving this statement immediately implies that *RSD* is characterized by equal treatment of equals, *ex post* efficiency, and strategyproofness. In case the statement does not hold, a counterexample might give us new insights and ideas to construct a counterexample for the original characterization. In particular, each counterexample of the original conjecture must also be a counterexample for Conjecture 1.

We now reformulate the problem as a system of linear equations such that every rule satisfying all axioms from Conjecture 1 is a solution to the system. As already mentioned, we can represent assignment rules f as vectors $\mathbf{x}$, where $\mathbf{x}_{(R,i,h)} = f(R,i,h)$ for all profiles R, agents i, and houses h. The constraints induced by the axioms are represented by the rows of a matrix $\mathbf{A}$ and a vector $\mathbf{b}$, such that $\mathbf{Ax} = \mathbf{b}$ if f represented by $\mathbf{x}$ satisfies all axioms. The columns of $\mathbf{A}$ correspond to the triples (R,i,h). Define $e_{(R,i,h)} \in \mathbb{R}^{1\times n^2 n!^n}$ as the unit vector with 1 at entry (R,i,h) and 0 otherwise. The rows of $\mathbf{A}$ have the following form depending on the type of axiom.

1. Bistochasticity constraints (excluding non-negativity constraints): $\mathbf{A}$ contains a row $\mathbf{a}_k$ for each profile R
   (a) and agent i, with $\mathbf{a}_k = \sum_{h\in H} e_{(R,i,h)}$, and
   (b) and house h, with $\mathbf{a}_k = \sum_{i\in N} e_{(R,i,h)}$.

   For such rows, $\mathbf{b}_k = 1$.

2. Support efficiency: $\mathbf{A}$ contains a row $\mathbf{a}_k$ for each triple (R, i, h) satisfying $RSD(R, i, h) = 0$, with $\mathbf{a}_k = e_{(R,i,h)}$. For such rows, $b_k = 0$.

3. Localizedness: $\mathbf{A}$ contains a row $\mathbf{a}_k$ for each profile R, agent i, house h, and each possible adjacent swap to profile R' that agent i can perform that does not move house h, with $\mathbf{a}_k = e_{(R,i,h)} - e_{(R',i,h)}$. For such rows, $\mathbf{b}_k = 0$.

4. Equal treatment of equals: $\mathbf{A}$ contains a row $\mathbf{a}_k$ for each profile R, house h, and agent pair (i, j) such that $i \neq j$ and $\succ_i = \succ_j$, with $\mathbf{a}_k = e_{(R,i,h)} - e_{(R,j,h)}$. For such rows, $\mathbf{b}_k = 0$.

As RSD satisfies all axioms, $\mathbf{A}\mathbf{x}^{RSD} = \mathbf{b}$, where $\mathbf{x}^{RSD}$ is the vector representing RSD. In general, it does not hold that every solution to $\mathbf{A}\mathbf{x} = \mathbf{b}$ corresponds to a valid assignment rule since the non-negativity of variables $\mathbf{x}_{(R,i,h)}$ with $\mathbf{x}^{RSD}_{(R,i,h)} \geq 0$ is not guaranteed. Nevertheless, the structure of RSD allows us to mix any other solution with $\mathbf{x}^{RSD}$ in a way that returns a new assignment rule satisfying all axioms.

Proposition 2. *Let $\mathbf{y} \neq \mathbf{x}^{RSD}$ be a solution to $\mathbf{A}\mathbf{x} = \mathbf{b}$. Then, there exists $\lambda > 0$ such that $\lambda\mathbf{y} + (1 - \lambda)\mathbf{x}^{RSD}$ is an assignment rule that satisfies all axioms and differs from RSD.*

Proof. Apart from non-negativity, $\lambda\mathbf{y} + (1 - \lambda)\mathbf{x}^{RSD}$ satisfies all axioms for all $\lambda \in [0, 1]$ as

$$\mathbf{A}(\lambda\mathbf{y} + (1 - \lambda)\mathbf{x}^{RSD}) = \lambda\mathbf{A}\mathbf{y} + (1 - \lambda)\mathbf{A}\mathbf{x}^{RSD} = b.$$

In order to ensure non-negativity, choose λ^* such that $\lambda^*\mathbf{y}_{(R,i,h)} + (1-\lambda^*)\mathbf{x}^{RSD}_{(R,i,h)} \geq 0$ for all (R, i, h). This is possible due to the fact that $\mathbf{x}^{RSD}_{(R,i,h)} = 0$ implies $\mathbf{y}_{(R,i,h)} = 0$ as $\mathbf{y}$ satisfies support efficiency.

Thus, $\lambda^*\mathbf{y} + (1 - \lambda^*)\mathbf{x}^{RSD}$ corresponds to an assignment rule that satisfies all axioms and differs from RSD since the representation of a rule in terms of $(\mathbf{x}_{(R,i,h)})$ is unique by definition. □

Proposition 2 shows that whenever there exists a solution $\mathbf{y} \neq \mathbf{x}^{RSD}$ to $\mathbf{A}\mathbf{x} = \mathbf{b}$, Conjecture 1 cannot hold. Furthermore, $\mathbf{y} - \mathbf{x}^{RSD} \neq \mathbf{0}$ lies in the kernel $\ker(\mathbf{A})$ of $\mathbf{A}$.

Corollary 1. *The following statements are equivalent:*

- *Conjecture 1 holds, i.e., the only solution to $\mathbf{A}\mathbf{x} = \mathbf{b}$ is $\mathbf{x}^{RSD}$.*
- *$\mathbf{A}$ has full rank, i.e., $rank(\mathbf{A}) = n^2 n!^n$.*
- $\ker(\mathbf{A}) = \{\mathbf{0}\}$.

In the next section, we use Proposition 2 and these equivalences to devise an algorithm that is able to solve Conjecture 1 for $n \leq 5$. We believe that Corollary 1 could also be helpful for finding a general analytic proof of Conjecture 1.

4 Checking whether the matrix has full rank

The following algorithm shows that Conjecture 1 holds for $n \leq 5$ by proving that $\mathbf{A}$ has full rank. Proposition 2 shows that this is equivalent to proving Conjecture 1 which in turn implies the original RSD characterization for $n \leq 5$. In principle, the rank of $\mathbf{A}$ can be computed using standard methods such as Gaussian elimination. However, there are two main issues with that approach. First, the size of the matrix is larger than $n^2 n!^n \times n^2 n!^n$. This can be partially mitigated because the matrix is sparse. Even though most entries are zero and do not need to be stored in memory, the remaining matrix is still very large. Second, standard methods often run into numerical problems.

To circumvent these issues, the algorithm we propose in this section uses search to construct all $n^2 n!^n$ rows $e_{(R,i,h)}$ using elementary row operations implying that the matrix has full rank. In particular, we add or substract multiples of one row from another or multiply a row by -1. Division is only used when we found a row that has only one non-zero entry to normalize. In this way the algorithm is guaranteed to not run into any numerical problems. Furthermore, we never explicitly construct the matrix, and use the symmetry of the domain to simplify the computation. This allows us to show that the matrix has full rank for $n \leq 5$.

The main idea of the algorithm builds on the fact that localizedness is the only axiom which connects profiles, i.e., the rows of matrix $\mathbf{A}$ have nonzero entries in different preference profiles. On the contrary, for all other axioms, the rows have nonzero entries only for a single profile. Starting with some preference profile R_s where all agents share the same preferences, it is possible to build the rows $e_{(R_s,i,h)}$ for all agent-house pairs (i, h) using elementary row operations. This can be done by adding "equal treatment of equals rows" to "bistochasticity rows" until the only nonzero entry is at index (R, i, h). With this method we can construct $e_{(R_s,i,h)}$ for all agent-house pairs (i, h). From an axiomatic point of view, it is clear that all agents need to receive the same assignment in R_s.

Next, the new rows $e_{(R_s,i,h)}$ can be added to the localizedness rows to build new rows $e_{(R',i,h)}$ for profiles R' that can be reached by swap manipulations from R_s. The algorithm can then try to solve these profiles, find new rows, and then propagate them further.

Thus, the algorithm consists of two parts, namely

- a subroutine that evaluates a single profile R and builds as many rows $e_{(R,i,h)}$ using elementary row operations as possible, and
- the main loop which builds rows for profiles that can be reached using localizedness and chooses the next profile to evaluate.

In contrast to R_s, note that in general, it is not possible to completely "solve" a profile at the first visit. Therefore, the main loop uses a priority queue to track which profile received the most rows since it was last considered. Guiding the search using this heuristic improves the runtime of the algorithm over naive breath first search or depth first search.

The algorithm continues the search until the identity matrix is contained in $\mathbf{A}$ or it proves that this is not possible. For that, it keeps track of the triples (R, i, h) for which the row $e_{(R,i,h)}$ was constructed with an indicator function $I_{RSD} : \mathcal{R} \times N \times H \to \{1, 0\}$ that returns 1 if the row $\mathbf{e}_{(R,i,h)}$ s already contained in the matrix and 0 otherwise. This indicator function is updated during program execution. When we refer to I_{RSD}, we refer to the current state of algorithm execution, unless stated otherwise. At the start of the algorithm $I_{RSD} \equiv 0$ is initialized to be 0 for every triple. In a first step, it sets $I_{RSD}(R, i, h) = 1$ for all triples (R, i, h) with $RSD(R, i, h) = 0$ since for those, $a_k = e_{(R,i,h)}$ by definition. Once $I_{RSD} \equiv 1$, the algorithm terminates as it has shown that the matrix $\mathbf{A}$ has full rank. We first present the subroutine, then the complete algorithm.

4.1 Solving single profiles

Given a preference profile R and indicator function I_{RSD}, the following subroutine computes all agent-house pairs (i, h) for which the vector $e_{(R,i,h)}$ can be constructed. We start by writing the rows corresponding to the bistochasticity and equal treatment of equals constraints of R into a separate matrix $\mathbf{B}$. The main idea then is to simplify these rows by setting all indices (R, i, h) to zero if $I_{RSD}(R, i, h) = 1$. This is allowed since $I_{RSD}(R, i, h) = 1$ implies

Algorithm 1 Subroutine that constructs new rows $e_{(R,i,h)}$ for input profile R.

Input
- R Preference profile
- I_{RSD} Function $I_{RSD} : \mathcal{R} \times N \times H \to \{0, 1\}$

```
B ← Matrix with bistochasticity and equal treatment of equals rows for profile R
while I_RSD was updated do
    while I_RSD was updated do
        for all (i, h) ∈ N × H do
            if RSD(R, i, h) = 1 then
                b_(i,h) ← 0 for all rows b in B
            end if
        end for
        for all Rows b in B do
            if ∃(i, h) ∈ N × H such that b = e_(R,i,h) and I_RSD(R, i, h) = 0 then
                I_RSD(R, i, h) ← 1.
            end if
        end for
    end while
    for all Rows b in B and h ∈ H do
        if ∀i ∈ N : b_(i,h) = 1 ⇒ ∀j ∈ N (b_(j,h) = 1 ⇔ ≻_i = ≻_j) then
            for all i ∈ N if b_(i,h) = 1 do
                I_RSD(R, i, h) ← 1.
            end for
        end if
    end for
end while
```

$e_{(R,i,h)}$ was constructed which in turn allows us to add or substract it from each row in $\mathbf{B}$ such that the entry becomes 0.

If the resulting matrix $\mathbf{B}$ contains rows with only one nonzero entry at position (R, i, h), then we set $I_{RSD}(R, i, h)$ to 1 and go back to the previous step. Otherwise, no simplifications are possible. We check if combining the resulting equal treatment of equals and bistochasticity rows results in new rows with only one nonzero entry. To do so, it is sufficient to check for each bistochasticity row b and house h if for all agents i with $b_{(i,h)} = 1$ we have for all agents j that $b_{(i,h)} = b_{(j,h)} = 1$ if and ony if $\succ_i = \succ_j$. If this is the case, we can construct the rows $e_{(R,i,h)}$ for all i for which $b_{(i,h)} = 1$ by adding the equal treatment of equals rows to the bistochasticity row. For these rows, the algorithm can once again set $I_{RSD}(R, i, h) = 1$ and go back to the first step. Otherwise, if no new rows are found, the subroutine terminates and returns the updated indicator function.

The subroutine only uses elementary row operations to construct new rows. Furthermore, it can restrict the matrix to a single profile R since it only considers matrix rows that have only zero entries for all indices of other profiles. Thus, these operations do not alter the rank of the matrix $\mathbf{A}$.

Another important property of the subroutine is that it is symmetric with respect to inputs. In other words, if we permute all inputs with some permutation of the agents $\pi \in \Pi$ and houses $\tau \in \mathcal{T}$, the updates to the indicator function are permuted by the same permutation. This property follows from the fact that the algorithm is deterministic and permutations of the profile permute the indices of the matrix $\mathbf{B}$ in the same way. Thus, the results are the same up to permutation.

4.2 Guided search and localizedness

Algorithm 1 is able to evaluate single profiles. All that is now left to do is to decide which profile to evaluate and to combine the new rows with localizedness. The full algorithm is described in Algorithm 2.

The first step is to initialize the indicator function I_{RSD} that keeps track of the rows $e_{(R,i,h)}$ that where already build. We initialize all entries with 0, except for the triples (R, i, h) with $RSD(R, i, h) = 0$.

Then, we use a standard best-first search algorithm to choose which profile to evaluate next. The heuristic used to determine the priority of profile R is the number of rows $e_{(R,i,h)}$ that where constructed since the last time the profile was considered. The priority queue is initialized with the profile R_s where all agents have the same preferences. This profile is a good choice since the submatrix of this profile has full rank and the bistochasticity and equal treatment of equals constraints are already sufficient to construct all $e_{(R_s,i,h)}$. Although we use a search algorithm, it has no "goal profile" in the usual sense but rather searches until it completed the indicator function or fails to do so. The advantage of best-first over depth-first or breath-first search is that it is much faster as it first evaluates profiles that are likely to be solved completely by the subroutine Algorithm 1. We observed other methods to visit the same profiles more frequently on average.

The algorithm then combines the rows found by the subroutine with localizedness by multiplying the localizedness row with -1 if necessary and adding the row from the subroutine. More precisely, if $I_{RSD}(R, i, h) = 1$ and agent i manipulates by rearranging houses above and below h, then $I_{RSD}(R', i, h) \leftarrow 1$, where R' is the profile agent i manipulates to. Therefore, we can set $I_{RSD}(R', i, h) = 1$ if $I_{RSD}(R, i, h) = 1$. We further reduce the number of manipulations that need to be considered by only allowing swap manipulations of adjacent houses. However, this does not really constitute a restriction since the same manipulations can be carried out by performing multiple swaps, i.e., all other manipulations are linearly dependent on pairwise swap rows.

This algorithm is still not efficient enough to solve the case of $n = 5$. In order to reduce the size of $\mathcal{R}$, we take advantage of the symmetry of the axioms and prove that the algorithm can assume symmetry without loss of generality. In particular, we show that the result of the algorithm on all canonical profiles $\mathcal{R}^*$ generalizes to $\mathcal{R}$ when ensuring that a manipulation that leaves the domain falls back to a canonical profile. For example, if agent i manipulates from profile $R \in \mathcal{R}^*$ to $R' \in \mathcal{R} \setminus \mathcal{R}^*$ then the algorithm assumes i manipulated from R to $canonical(R', i)$, where *canonical* is a function that maps a profile to the canonical profile.

A very important detail here is that while this function always maps to a single profile, the manipulating agent i might map to multiple agents in the canonical profile. To account for this we let the function *canonical* also return a list of agents in the new profile that the manipulator can map to. If Algorithm 2 is used on $\mathcal{R}$, *canonical* simply returns the corresponding profile and agent.

Lemma 1. *The result of Algorithm 2 holds for $\mathcal{R}$ when the search space is restricted to $\mathcal{R}^*$.*

Proof. We show that Algorithm 2 on $\mathcal{R}^*$ is equivalent to Algorithm 2 on $\mathcal{R}$ by induction. Let $I_{RSD} : \mathcal{R} \times N \times H \to \{0, 1\}$ and $I^*_{RSD} : \mathcal{R}^* \times N \times H \to \{0, 1\}$ be the indicator functions for the first and second program, respectively. Denote Π as the set of all permutations of agents and $\mathcal{T}$ as the set of all permutations of the houses, i.e., $\pi \in \Pi$, $\tau \in \mathcal{T}$ maps $\pi(\tau(R)) = R'$ a preference profile R to another preference profile R' by rearranging the agents according to a permutation π and renaming the houses according to a permutation τ. Obviously, $|\Pi| = |\mathcal{T}| = n!$ as both sets consists of $n!$ permutations of the agents and houses, respectively. Our induction proof is based on the idea that Algorithm 2 on $\mathcal{R}$ will, after some extra steps, return to a state that is equivalent to Algorithm 2 on $\mathcal{R}^*$. We show this by induction over the outermost loop of Algorithm 2. In particular, we show that there exists an execution of Algorithm 2 on $\mathcal{R}$ such that the following invariance holds at some point.

$$I^*_{RSD}(R, i, h) = I_{RSD}(\pi(\tau(R)), \pi(i), \tau(h)) \quad \forall R \in \mathcal{R}^*, \pi \in \Pi, \tau \in \mathcal{T}, i \in N, h \in H \qquad (1)$$

Induction base: At the start of the algorithm, $I_{RSD} = I^*_{RSD} \equiv 0$ meaning that the induction hypothesis trivially holds. It still holds after the support efficiency constraints are added to I_{RSD} since *RSD* satisfies symmetry.

Induction hypothesis: Equation (1) holds at the start of the k-th iteration of the outermost loop.

Induction step: We show Equation (1) holds at the end of the k-th iteration of the outermost loop. Algorithm 2 will look at profile $R \in \mathcal{R}^*$ in the k-th iteration. Let the variant on $\mathcal{R}$ look at all profiles in $[R]$ which denotes the equivalence class of all profiles equivalent to R by symmetry. Clearly, both algorithms do not change the indicator value of any profile that is not in $[R]$ or a neighbor of it. In line 12, the algorithm calls the subroutine.

The subroutine Algorithm 1 is deterministic and permutations of the inputs result in the same permutations of the outputs implies that since the second program permutes the inputs, the outputs are also permuted. If the second program sets $I^*_{RSD}(R, i, h) = 1$, the first program is able to set $I_{RSD}(\pi(\tau(R)), \pi(i), \tau(h)) = 1$ for every $\pi \in \Pi, \tau \in \mathcal{T}$ by induction hypothesis. Therefore, the invariance condition is preserved for profiles in $[R]$.

Next, in line 13, the algorithm starts to iterate over neighbors of R that can be reached by adjacent swap manipulations of the agents. Let R' be an arbitrary neighboring profile, i the manipulating agent, and $k \in [n-1]$ the position in agent i's preferences such that for all $j \neq i$, the preferences stay the same ($\succ_i = \succ'_j$) and $\succ'_i = \mathit{swap}(\succ_i, k, k+1)$. Furthermore, let $R'' = \mathit{canonical}(R')$ be the canonical representation of R' and $\pi' \in \Pi$, $\tau' \in \mathcal{T}$ be any pair of permutations that maps R'' to R'. For each $l \in [n] \setminus \{k, k+1\}$ the algorithm performs the following operations. Let h be agent i's lth most preferred house. Then, if $I_{RSD}(R, i, h) = 1$ and $I_{RSD}(R'', i, h) = 0$, set $I_{RSD}(R'', i, h) \leftarrow 1$. This is allowed since the localizedness row together with $e_{(R,i,h)}$ and multiplication by -1 if necessary can reach $e_{(R'',i,h)}$. The first program performs the same operation but for each profile in $[R]$. By induction hypothesis, $I^*_{RSD}(R, i, h) = I_{RSD}(\pi(\tau(R)), \pi(i), \tau(h))$ and $I^*_{RSD}(R'', i, h) = I_{RSD}(\pi(\tau(R'')), \pi(i), \tau(h))$ for all permutations $\pi \in \Pi$ and $\tau \in \mathcal{T}$. Thus, the condition of the if statement $I_{RSD}(R, i, h) = 1$ and $I_{RSD}(R'', i, h) = 0$ is true in the second program if and only if it is true in the first program for each permutation π, τ. Consequently, $I^*_{RSD}(R'', i, h) = I_{RSD}(\pi(\tau(R'')), \pi(i), \tau(h)) \leftarrow 1$ for all permutations π and τ. Again the induction hypothesis is preserved. Since no other operations change the indicator function, we conclude that the invariance holds after each step of Algorithm 2. □

To summarize, it is sufficient to restrict the algorithm to $\mathcal{R}^*$ and all actions of the algorithm can be represented as elementary row operations. As they do not change the rank of a matrix and the algorithm shows that the full identity matrix can be constructed from the matrix $\mathbf{A}$, we conclude that $\mathbf{A}$ has full rank. Corollary 1 then implies that Conjecture 1 holds. We ran the algorithm successfully for all $n \leq 5$.

5 Conclusion

The current state of *RSD* characterizations via equal treatment of equals, *ex post* efficiency, and strategyproofness for small n are summarized in Figure 1. The first characterization for $n = 3$ was shown by Bogomolnaia and Moulin [3]. In their proof, they use a lemma that

Algorithm 2 Verify *RSD* Characterization

Input

n Number of Agents and Objects

$I_{RSD} \leftarrow 0$ ▷ Initialize $I_{RSD} : \mathcal{R}^* \times N \times U \rightarrow \{1, 0\}$ as the constant 0 function.
for all $(R, i, h) \in \mathcal{R} \times N \times H$ **do**
  if $RSD(R, i, h) = 0$ **then** ▷ $f(R, i, h) = 0$ due to support efficiency.
    $I_{RSD}(R, i, h) \leftarrow 1$
  end if
end for
$queue \leftarrow$ **new** *Priority Queue*
$queue.insert(R_s, 0)$
while *queue* **is not** *empty* **do**
  $R \leftarrow queue.findmax()$
  $queue.deletemax()$
  Algorithm 1(R, I_{RSD}) ▷ Algorithm 1 updates I_{RSD}.
  for all R' s.t. $\exists i \in N\ \forall j \neq i\ \ \succ_j = \succ'_j \wedge \exists k \in [n]\ \ \succ'_i = swap(\succ_i, k, k+1)$ **do**
    $R^*, manipulators = canonical(R', i)$
    $\Delta \leftarrow 0$
    for all $l \in [n] \setminus \{k, k+1\}$ **do**
      for all $i^* \in manipulators$ **do**
        $h \leftarrow lth\ best(\succ_i, l)$
        $h^* \leftarrow lth\ best(\succ^*_{i^*}, l)$
        if $I_{RSD}(R, i, h) = 1$ **and** $I_{RSD}(R^*, i^*, h^*) \neq 0$ **then**
          $I_{RSD}(R^*, i^*, h^*) \leftarrow 1$
          $\Delta \leftarrow \Delta + 1$
        end if
      end for
    end for
    if $\Delta > 0$ **then**
      if $R' \in queue$ **then**
        $queue.increasepriority(R^*, \Delta)$
      else
        $queue.insert(R^*, \Delta)$
      end if
    end if
  end for
end while
return $I_{RSD} \equiv 1$ ▷ The characterization holds if I_{RSD} equals 1 for every (R, i, h).

	Extra Condition	Strategyproofness	Source
$n \leq 3$	—	strategyproofness	Bogomolnaia and Moulin [3]
$n \leq 4$	symmetry	only localizedness	Sandomirskiy [19]
$n \leq 5$	—	only localizedness	this paper

Figure 1: Overview of characterizations of *RSD* via equal treatment of equals, support efficiency, and strategyproofness for small n. It is open whether *ex post* efficiency and nonperverseness are required for larger n.

is based on a weakening of support efficiency. Since *ex post* efficiency and support efficiency are equivalent for $n = 3$, full *ex post* efficiency is not required for $n = 3$.

Recently, Sandomirskiy [19] has shown via a computer-aided proof that the characterization holds for $n \leq 4$ using symmetry, support efficiency, and localizedness. We extend this result by showing that the *RSD* characterization holds for $n \leq 5$ even when replacing symmetry with equal treatment of equals. This raises the question whether support efficiency and localizedness also suffice for arbitrary n.

When analyzing the arguments produced by our algorithm, it turns out that certain profiles require very long chains of reasoning that argue over many other profiles across the full domain. In particular, it does not seem possible to partition $\mathcal{R}^*$ by, e.g., first looking at all profiles where every agent top-ranks the same house and then reuse results for smaller n.

As a consequence, we suspect that the characterization cannot hold in many subdomains of $\mathcal{R}$. As an example, consider the subdomain $\mathcal{R}^>$ where all agents have the same ranking over all houses but one, introduced by Chang and Chun [6]. This domain is rich enough for the impossibility of equal treatment of equals, strategyproofness, and ordinal efficiency by Bogomolnaia and Moulin [3]. In this domain, *RSD* is *not* the only rule satisfying equal treatment of equals, strategyproofness, and *ex post* efficiency for $n = 4$. An alternative rule was found using quadratic programming and has the property that it has the maximal $L2$-distance to *RSD* when considering the summed distance over all profiles. Furthermore, it satisfies symmetry on the subdomain, profiles that are in the same equivalence class as given profiles have the same assignment permuted accordingly.

Theorem 1. *RSD is not characterized by equal treatment of equals,* ex post *efficiency, and strategyproofness in the domain $\mathcal{R}^>$.*

It remains an open problem whether a characterization of *RSD* via *ex post* efficiency, strategyproofness and equal treatment of equals holds for arbitrary n. On the one hand, our results suggest that such a characterization might indeed hold, even when weakening efficiency and strategyproofness and without additionally demanding symmetry. In fact, the weaker axioms, in particular support efficiency instead of *ex post* efficiency, seem to be a lot easier to handle for computers as well as humans. On the other hand, in case the characterization does not hold, our results show that another *ex post* efficient and strategyproof rule that treats equals equally can only differ from *RSD* when $n \geq 6$, casting doubt on the existence of a closed-form representation of any such rule.

We hope that the linear algebraic interpretation of the problem presented in this paper will prove beneficial for a complete characterization of *RSD*.

Acknowledgments

This work was supported by the Deutsche Forschungsgemeinschaft under grants BR 2312/11-2 and BR 2312/12-1. We thank Florian Brandl, Marek Pycia, Fedor Sandomirskiy, and Omer Tamuz for helpful discussions.

References

[1] A. Abdulkadiroğlu and T. Sönmez. Random serial dictatorship and the core from random endowments in house allocation problems. *Econometrica*, 66(3):689–701, 1998.

[2] H. Aziz, F. Brandt, and M. Brill. The computational complexity of random serial dictatorship. *Economics Letters*, 121(3):341–345, 2013.

[3] A. Bogomolnaia and H. Moulin. A new solution to the random assignment problem. *Journal of Economic Theory*, 100(2):295–328, 2001.

[4] S. Bouveret, Y. Chevaleyre, and N. Maudet. Fair allocation of indivisible goods. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 12. Cambridge University Press, 2016.

[5] F. Brandt. Rolling the dice: Recent results in probabilistic social choice. In U. Endriss, editor, *Trends in Computational Social Choice*, chapter 1, pages 3–26. AI Access, 2017.

[6] H.-I. Chang and Y. Chun. Probabilistic assignment of indivisible objects when agents have the same preferences except the ordinal ranking of one object. *Mathematical Social Sciences*, 90:80–82, 2017.

[7] Y. Chevaleyre, P. E. Dunne, U. Endriss, J. Lang, M. Lemaître, N. Maudet, J. Padget, S. Phelps, J. A. Rodríguez-Aguilar, and P. Sousa. Issues in multiagent resource allocation. *Informatica*, 30:3–31, 2006.

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

[9] D. E. Knuth. An exact analysis of stable allocation. *Journal of Algorithms*, 20:431–442, 1996.

[10] M. Manea. Serial dictatorship and Pareto optimality. *Games and Economic Behavior*, 61:316–330, 2007.

[11] D. F. Manlove. *Algorithmics of Matching Under Preferences*. World Scientific Publishing Company, 2013.

[12] T. Mennle and S. Seuken. Partial strategyproofness: Relaxing strategyproofness for the random assignment problem. *Journal of Economic Theory*, 191:105–144, 2021.

[13] S. Pápai. Strategyproof assignment by hierarchical exchange. *Econometrica*, 68(6): 1403–1434, 2000.

[14] D. C. Parkes and S. Seuken. *Economics and Computation*. Cambridge University Press, Forthcoming.

[15] M. Pycia and P. Troyan. Strategy-proof, efficient, and fair allocation: Beyond random priority. 2023. Working paper.

[16] M. Pycia and P. Troyan. A theory of simplicity in games and mechanism design. *Econometrica*, 2023. Forthcoming.

[17] M. Pycia and M. U. Ünver. Incentive compatible allocation and exchange of discrete resources. *Theoretical Economics*, 12(1):287–329, 2017.

[18] D. Saban and J. Sethuraman. The complexity of computing the random priority allocation matrix. *Mathematics of Operations Research*, 40(4):1005 –1014, 2015.

[19] F. Sandomirskiy. Private Communication, July 2022.

[20] T. Sönmez and M. U. Ünver. Matching, allocation, and exchange of discrete resources. In J. Benhabib, M. O. Jackson, and A. Bisin, editors, *Handbook of Social Economics*, volume 1, chapter 17, pages 781–852. Elsevier, 2011.

[21] L.-G. Svensson. Strategy-proof allocation of indivisible goods. *Social Choice and Welfare*, 16(4):557–567, 1999.

[22] W. S. Zwicker. Introduction to the theory of voting. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 2. Cambridge University Press, 2016.

A Omitted Proofs

In the appendix, we provide the missing proofs of the main body. We start with the claims from the preliminaries.

Proposition 1. *Every symmetric assignment rule satisfies equal treatment of equals.*

Proof. Let f be a symmetric assignment rule and R be an arbitrary profile with $\succ_i = \succ_j$ for two agents $i, j \in N$. Consider the permutation $\pi = (ij)$ that only swaps the identities of agents i and j. As $\succ_i = \succ_j$, $R = \pi \circ R$ implies $f(R) = \pi \circ f(R)$ by symmetry. In particular, $f(R, i) = \pi \circ f(R, i) = f(R, j)$ showing that agents i and j receive the same assignment under f in R. □

Continuing on, we prove the claim that support efficiency and *ex post* efficiency are equivalent for $n \leq 3$. Example 1 shows that this is no longer the case when $n \geq 4$.

Proposition 3. *Support efficiency and* ex post *efficiency coincide for $n \leq 3$.*

Proof. The case $n = 2$ is easily solved by exhausting all cases. If the two agents disagree on their top choice, only one deterministic assignment is efficient. Therefore all assignments that violate *ex post* efficiency also violate support efficiency. Otherwise, the two agents share the same preferences, in this case all random assignments are *ex post* efficient and thus also support efficient.

For the case $n = 3$, assume that a preference profile R and random assignment $f(R)$ exist such that $f(R)$ is support efficient but not *ex post* efficient. Then, there exists a deterministic assignment M that is not efficient that is needed to represent $f(R)$. Furthermore, by support efficiency, the support of M is efficient.

We consider two cases. M can be made efficient either by letting three agents trade their houses in a circular fashion, or by swapping the houses of two agents. In the first case, M is obviously not support efficient as all three agents improve, meaning that no agent received her top choice in M. For the second case, two agents, w.l.o.g. 1 and 2, both improve when they swap houses h_1 and h_2, i.e., $h_1 \succ_1 h_2$ and $h_2 \succ_2 h_1$ but 1 receives h_2 and 2 receives h_1 in M. Assume now, again w.l.o.g., that $h_1 \succ_3 h_2$. It is obvious that in this case agent 2 cannot receive h_1 in any efficient deterministic assignment. Again, M violates support efficiency.

We have shown that for $n = 3$, a violation of *ex post* efficiency implies a violation of support efficiency. Since *ex post* efficiency implies support efficiency, they are equivalent for $n = 3$. □

Finally, we state the proof of Theorem 1.

					h_1	h_2	h_3	h_4
1 :	h_1	h_2	h_3	h_4	$\frac{3}{4}$	0	$\frac{1}{24}$	$\frac{5}{24}$
2 :	h_2	h_1	h_3	h_4	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{6}$	$\frac{1}{4}$
3 :	h_2	h_3	h_1	h_4	0	$\frac{1}{3}$	$\frac{5}{12}$	$\frac{1}{4}$
4 :	h_2	h_3	h_4	h_1	0	$\frac{1}{3}$	$\frac{3}{8}$	$\frac{7}{24}$

					h_1	h_2	h_3	h_4
1 :	h_1	h_2	h_3	h_4	$\frac{3}{4}$	0	0	$\frac{1}{4}$
2 :	h_2	h_1	h_3	h_4	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{6}$	$\frac{1}{4}$
3 :	h_2	h_3	h_4	h_1	0	$\frac{1}{3}$	$\frac{5}{12}$	$\frac{1}{4}$
4 :	h_2	h_3	h_4	h_1	0	$\frac{1}{3}$	$\frac{5}{12}$	$\frac{1}{4}$

					h_1	h_2	h_3	h_4
1 :	h_2	h_1	h_3	h_4	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{24}$	$\frac{5}{24}$
2 :	h_2	h_1	h_3	h_4	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{24}$	$\frac{5}{24}$
3 :	h_2	h_3	h_1	h_4	0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$
4 :	h_2	h_3	h_4	h_1	0	$\frac{1}{4}$	$\frac{5}{12}$	$\frac{1}{3}$

					h_1	h_2	h_3	h_4
1 :	h_2	h_1	h_3	h_4	$\frac{1}{2}$	$\frac{1}{4}$	0	$\frac{1}{4}$
2 :	h_2	h_1	h_3	h_4	$\frac{1}{2}$	$\frac{1}{4}$	0	$\frac{1}{4}$
3 :	h_2	h_3	h_4	h_1	0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$
4 :	h_2	h_3	h_4	h_1	0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$

					h_1	h_2	h_3	h_4
1 :	h_2	h_1	h_3	h_4	$\frac{2}{3}$	$\frac{1}{4}$	0	$\frac{1}{12}$
2 :	h_2	h_3	h_1	h_4	$\frac{1}{6}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{4}$
3 :	h_2	h_3	h_4	h_1	$\frac{1}{12}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{3}$
4 :	h_2	h_3	h_4	h_1	$\frac{1}{12}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{3}$

Figure 2: The five canonical profiles are the only canonical profiles for which the proposed rule returns a different output than *RSD*. Furthermore, only entries marked in gray differ from *RSD*. The rule also satisfies symmetry within domain $\mathcal{R}^{>}$.

Theorem 1. *RSD is not characterized by equal treatment of equals,* ex post *efficiency, and strategyproofness in the domain* $\mathcal{R}^{>}$.

Proof. The rule defined in Figure 2 satisfies all three axioms and was found using quadratic programming. It is equal to RSD on all canonical profiles except the five shown in Figure 2. Profiles in the same equivalence class receive the same random assignment permuted accordingly. □

Contact Details

Felix Brandt
Technische Universität München
München, Germany
Email: `brandt@tum.de`

Matthias Greger
Technische Universität München
München, Germany
Email: `matthias.greger@tum.de`

René Romen
Technische Universität München
München, Germany
Email: `rene.romen@tum.de`