Characterizing Random Serial Dictatorship

Felix Brandt Matthias Greger René Romen
Technische Universität München

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 [4] have shown that *RSD* is characterized by these three axioms. Using best first search in combination with quadratic programming, we extend this characterization to $n \leq 5$, getting closer to answering the long-standing open question whether the characterization holds for arbitrary n. On the way, we describe weakenings of *ex post* efficiency and strategyproofness that are sufficient for our characterization and identify problems when making statements for larger n.

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., 8, 19, 12, 5]. 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 [4].

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, 10]. Svensson [20] showed that any deterministic, strategyproof, nonbossy, and neutral mechanism is serially dictatorial, implying that the symmetrization of such a rule has to coincide with *RSD*. Replacing neutrality with efficiency, Bade [3], who calls any rule that satisfies these axioms *good*, characterizes the set of all good rules as trading and braiding mechanisms and proves that the symmetrization of any of these rules still coincides with *RSD*. Interestingly, she also raises the question whether the statement still holds without nonbossiness and points out that this would be a direct consequence of *RSD* being characterized by *ex post* efficiency, strategyproofness and equal treatment of equals (agents with the same preferences should receive the same probabilistic assignment).

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 [4]. 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 clear any doubts on its optimality in settings where strategyproofness is indispensable.

Unfortunately, to the best of our knowledge, there does not even exist a characterization of all deterministic, strategyproof, efficient and neutral mechanisms [cf. 20], which makes it very difficult to apply the techniques used by Bade. Furthermore, Aziz et al. [2] and Saban and Sethuraman [17] showed that it is NP-complete to decide whether an agent receives a given houses 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 mechanisms 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 computational approach to show that the desired characterization holds for $n \leq 5$. After introducing the necessary notation and central axioms in Section 2, we describe the algorithm for showing the characterization and prove its correctness in Section 3. Afterwards, we summarize and interpret the results obtained from the algorithm and surmise how a proof for arbitrary n might look like in Section 4.

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 tuples (i, h) for which $M_{i,h} > 0$. Whenever $M_{i,h} \in \{0, 1\}$, 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 the serial dictatorship mechanism 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 [11]. 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 for our results. 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 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 [4], 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]

[1]This version of strategyproofness for probabilistic assignment rules is sometimes also called (strong) *SD*-strategyproofness [see, e.g., 6].

Definition 3. An assignment rule 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 $h \in H$.

To implement strategyproofness, we leverage a result from Gibbard [9], showing 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 localizedness is sufficient for all our results.

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 mechanism 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., 21]. 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.

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

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, $f(R', 1) = (1, 0)$ for $R' = (h_1 \succ_1 h_2, h_2 \succ_2 h_1)$, and $f(R'', 1) = (1, 0)$ for $R'' = (h_2 \succ_1 h_1, h_1 \succ_2 h_2)$. 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 $\tau \circ f(R') = f(\tau \circ R') = f(R'')$.

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 $|\mathcal{R}| = n!^n$.

[2] Gibbard considers the general social choice domain. Mennle and Seuken [13] have rediscovered this equivalence in the context of random assignment.

3 The algorithm

In this section, we present the algorithm used to show our RSD characterization results. To simplify notation, let f denote any assignment rule that satisfies equal treatment of equals, *ex post* efficiency, and strategyproofness. In particular, $f(R,i,h) = RSD(R,i,h)$ means that for all rules f that satisfy the three axioms, the equation holds, i.e., all rules that satisfy the three axioms are equivalent to RSD for this profile, agent, and house.

The algorithm consists of two main parts. The first part consists of a subroutine (Algorithm 1) which conducts evaluations on single profiles. Given a profile R, it computes for which agent-house pairs (i,h) the equation $f(R,i,h) = RSD(R,i,h)$ holds. Since this subroutine only considers a single profile, it uses equal treatment of equals, *ex post* efficiency, and constraints from the bistochastic assignment matrix, but not strategyproofness. Additionally, it takes into consideration for which pairs of agents and houses equivalence of all rules to RSD has already been shown for this profile.

The second part of the algorithm is a guided search that decides which profile to evaluate next with the subroutine. The results are then propagated to nearby profiles by swap manipulations of single agents.

This process is repeated until it either terminates with $f(R,i,h) = RSD(R,i,h)$ for all profiles R, agents i, and houses h or it decides that reaching this conclusion is not possible. In the first case, it is clear that the RSD characterization holds, while in the second case, no conclusive decision is possible because the algorithm only uses localizedness (and not nonperverseness).

The goal of the algorithm is to prove $f(R,i,h) = RSD(R,i,h)$ for each triple R,i,h. To track the triples for which $f(R,i,h) = RSD(R,i,h)$ has already been shown, the algorithm uses an indicator function $I_{RSD} : \mathcal{R} \times N \times H \to \{1,0\}$ that returns 1 if $f(R,i,h) = RSD(R,i,h)$ 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. Once $I_{RSD} \equiv 1$, the algorithm terminates as it has shown that the RSD characterization holds. We first present the subroutine, then the complete algorithm.

3.1 Evaluating a profile with quadratic programming

Here, we present the subroutine that, given a preference profile, computes all agent-house pairs for which the equation $f(R,i,h) = RSD(R,i,h)$ holds given the current indicator function, *ex post* efficiency, equal treatment of equals, and assignment constraints. The main idea is that a quadratic program (QP) is used to compute an assignment that has maximal $L2$-distance from the assignment returned by RSD while still satisfying all constraints, i.e., the $L2$-norm of the difference between the two assignments is maximized. From this assignment, we can extract agent-house pairs that receive probability different from RSD. We can then drop these pairs from the objective, i.e., we no longer consider them when measuring the distance and recourse. At some point the objective value becomes 0. Now, for all agent-house pairs that remain in the objective, $f(R,i,h) = RSD(R,i,h)$ holds, otherwise an assignment with $L2$-distance strictly larger than 0 exists, contradicting the objective value of 0. This set of agent-house pairs is then returned by the subroutine.

We now describe the QP in detail which is presented in Figure 1.

It receives as input a preference profile R, the indicator function I_{RSD}, and a set E of excluded agent-house pairs. The indicator function enables the QP to access the agent-house pairs for which $f(R,i,h) = RSD(R,i,h)$ was shown to hold. The set E on the other hand contains already excluded agent-house pairs for which the algorithm knows that there is currently not enough information to show $f(R,i,h) = RSD(R,i,h)$. The QP has variables

$$
\begin{array}{lll}
\max & \displaystyle\sum_{(i,h)\in N\times H\setminus E} (p_{i,h} - RSD(R,i,h))^2 & \\
\text{subject to} & p_{i,h} = RSD(R,i,h) & \forall i\in N, h\in H \text{ if } I_{RSD}(R,i,h)=1 \\
& p_{i,h} = p_{j,h} & \forall i,j\in N, h\in H \text{ if } \succ_i = \succ_j \\
& \displaystyle\sum_{\pi\in\Pi} \lambda_\pi SD(R,\pi,i,h) = p_{i,h} & \forall i\in N, h\in H \\
& \displaystyle\sum_{\pi\in\Pi} \lambda_\pi = 1 & \\
& \sum_{i\in N} p_{i,h} = 1 & \forall h\in H \\
& \sum_{h\in H} p_{i,h} = 1 & \forall i\in N \\
& \lambda_\pi \geq 0 & \forall \pi\in\Pi \\
& p_{i,h} \geq 0 & \forall i\in N, h\in H
\end{array}
$$

Figure 1: QP that finds the assignment with maximal $L2$-distance to $RSD(R)$ for profile R.

$p_{i,h}$ for all agents i and houses h and variables λ_π for each permutation of the agents $\pi \in \Pi$.

The objective is to maximize the $L2$-distance between the variables $p_{i,h}$ and $RSD(R,i,h)$, restricted to pairs $(i,h) \notin E$. The constraints are all linear. The first set of constraints ensures that $p_{i,h} = p_{j,h}$ if $I_{RSD}(R,i,h) = 1$. They enforce that $p_{i,h}$ is equal to RSD if the program has already proven that this is the case. The second set of constraints encodes equal treatment of equals, i.e., if $\succ_i = \succ_j$, then $p_{i,h} = p_{j,h}$ for all $h \in H$. The third set of constraints encodes *ex post* efficiency by enforcing that the resulting assignment can be represented as a convex combination of serial dictatorships $\sum_{\pi\in\Pi} \lambda_\pi SD(R,\pi) = p$. The remaining constraints ensure that all variables are non-negative and form a valid assignment. Note that these constraints are already induced by the efficiency constraints as serial dictatorships are valid assignment rules. We now prove that if the optimal objective value of the QP is 0, then $f(R,i,h) = RSD(R,i,h)$ holds for all pairs (i,h) that remain in the objective. Remember that the set E contains agent-house pairs that are excluded from the objective because it was determined that there is not yet enough information to prove $f(R,i,h) = RSD(R,i,h)$ at that point.

Lemma 1. *Let R be a profile, E a set of agent-house pairs, and I_{RSD} the current state of the indicator function during program execution, i.e., $I_{RSD}(R,i,h) = 1$ implies $f(R,i,h) = RSD(R,i,h)$. Then, for all $(i,h) \in N \times H \setminus E$, $f(R,i,h) = RSD(R,i,h)$ when the optimal objective value of the QP in Figure 1 is 0.*

The QP has another important property as a result of the symmetry of constraints that are independent of the input. When the inputs are permuted by a permutation of the agents $\pi \in \Pi$ and the houses $\tau \in \mathcal{T}$ then the optimal solutions are permuted by the same permutations. Let $R^{\pi,\tau} = \pi(\tau(R))$ be the profile where the agents in R are reordered according to π and the houses renamed according to τ. Furthermore, $I_{RSD}^{\pi,\tau} = \pi(\tau(I_{RSD}))$ is the permuted indicator function $I_{RSD}^{\pi,\tau}(R,i,h) = 1$ iff $I_{RSD}(R^{\pi,\tau}, \pi(i), \tau(h)) = 1$. The permutation of the set E is $E^{\pi,\tau} = \pi(\tau(E)) = \{(\pi(i), \tau(h)) | (i,h) \in E\}$ and the permutation of an assignment p is $p^{\pi,\tau} = \pi(\tau(p))$, i.e., $p_{i,h} = p^{\pi,\tau}_{\pi(i),\tau(h)}$. For space reasons we moved the proof of Lemmas 1 and 2 to the appendix.

Lemma 2. *Let $\pi \in \Pi$ be a permutation of the agents and $\tau \in \mathcal{T}$ a permutation of the*

Algorithm 1 QP

Input

R Preference profile

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

$E \leftarrow \emptyset$ ▷ Tracks pairs that can be unequal to RSD.
while True do
  $QP \leftarrow$ **new** $QP(R, I_{RSD}, E)$
  $QP.optimize()$
  if $QP.objectiveValue() = 0$ **then**
    return $\{(i, h) \in N \times H | (i, h) \notin E \wedge I_{RSD}(R, i, h) = 0\}$
  end if
  for all $(i, h) \in N \times H \setminus E$ **do**
    if $p_{i,h} \neq RSD(R, i, h)$ **then**
      $E \leftarrow E \cup \{(i, h)\}$
    end if
  end for
end while

houses. If p, λ is an optimal solution for $QP(R, I_{RSD}, E)$, then $p^{\pi,\tau}$, $\pi(\lambda)$ is an optimal solution for $QP(R^{\pi,\tau}, I^{\pi,\tau}_{RSD}, E^{\pi,\tau})$ with the same objective value.

The QP on its own is not sufficient to evaluate a profile because the goal is to find all agent-house pairs for which all rules are equal to RSD. The reason is that an assignment with maximal $L2$-distance to RSD does not necessarily differ on all entries that can be unequal to RSD in any assignment. We therefore use Algorithm 1 to find all such agent-house pairs. Algorithm 1 calls the QP multiple times until the optimal objective becomes 0. Each time the QP is executed and the optimal objective is greater than 0, at least four agent-house pairs are shown to be unequal to RSD. These pairs are added to a initially empty set E and used as input for the next time the QP is executed. Since the number of terms in the objective decreases in each iteration the algorithm is guaranteed to terminate. When it terminates, the optimal objective is 0 and according to Lemma 1, the algorithm has proven $f(R, i, h) = RSD(R, i, h)$ for all agent-house pairs that are still part of the objective, i.e., not in E. The indicator function I_{RSD} is then updated by the main algorithm to reflect these new insights. Finally, because of the symmetry of the QP shown in Lemma 2, Algorithm 1 is also symmetric in the sense that if the inputs are permuted, the outputs are permuted by the same permutations.

Our results imply that support efficiency is already sufficient to show the RSD characterization for $n \leq 5$. There are two ways to change our algorithm for this purpose. First, the QP can be updated to use support efficiency instead of *ex post* efficiency. This can be done by removing the *ex post* efficiency constraints and variables and replacing them with the constraints $p_{i,h} = 0$ if $RSD(R, i, h) = 0$ for all $i \in N, h \in N$. Second, the QP can be replaced by a simple algorithm that attempts to propagate 1's in the indicator function using equal treatment of equals, support efficiency, and assignment constraints. For the first variant, it is easy to see that Lemmas 1 and 2 still hold.

3.2 Propagating results with strategyproofness

We now integrate Algorithm 1 from the last section into Algorithm 2. The goal of the algorithm is to prove for each profile R, agent i and house h that $f(R, i, h) = RSD(R, i, h)$. The algorithm defines an indicator function $I_{RSD} : \mathcal{R} \times N \times H \to \{0, 1\}$ to keep track of the triples for which the equality was proven, i.e., the algorithm ensures that $I_{RSD}(R, i, h) = 1 \implies f(R, i, h) = RSD(R, i, h)$ always holds. At the start of the algorithm, the indicator function is initialized as the constant zero function $I_{RSD} \equiv 0$. It then uses Algorithm 1 to find new triples for which the equation holds and propagates these results to neighboring profiles using (parts of) strategyproofness. We describe this process in detail later in this section. First, we talk about how the size of $\mathcal{R}$ can be reduced. We take advantage of the symmetry of all axioms and show that the algorithm can assume symmetry without loss of generality. In particular, we show that the result of the algorithm on $\mathcal{R}^*$ generalizes to $\mathcal{R}$.

Lemma 3. *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 "equal to RSD" indicator functions for the first and second algorithm, respectively. Remember Π is the set of all permutations of agents and $\mathcal{T}$ 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 to another preference profile by rearranging the agents according to a permutation π from N to N and renaming the houses according to a permutation τ from H to H. Obviously, $|\Pi| = |\mathcal{T}| = n!$ because both consists of one 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.

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

Induction base: At the start of the algorithm, $I_{RSD} = I^*_{RSD} \equiv 0$ meaning that the induction hypothesis trivially holds.

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 $\{R' | R' \in [R]\}$ where $[R]$ 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. First, consider the changes to entries for profiles in $[R]$. In line 7, the algorithm calls the quadratic program. Lemma 2 implies that since the second program permutes the inputs, the outputs are also permuted. If the first program sets $I_{RSD}(R, i, h) = 1$, then the second program sets $I^*_{RSD}(\pi(\tau(R)), \pi(i), \tau(h)) = 1$. Therefore, the invariance condition is preserved for profiles in $[R]$.

Next, in line 11, the algorithm starts to iterate over neighbors of R that can be reached by adjacent swap manipulations of the agents. Let R' be the 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 $I_{RSD}(R'', i, h) \leftarrow 1$. The second algorithm 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 first program if and only if it is true in the second program for each permutation π. 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. □

We now explain Algorithm 2 in detail. As mentioned before, the algorithm defines an indicator function $I_{RSD} : \mathcal{R}^* \times N \times H \to \{0,1\}$ and ensures that $I_{RSD}(R,i,h) = 1 \implies f(R,i,h) = RSD(R,i,h)$ always holds. Due to Lemma 3, we replace $\mathcal{R}$ with $\mathcal{R}^*$ in the definition of I_{RSD}.

The algorithm also initiates a priority queue and uses it to determine which profile should be considered next. The priority that profiles receive is the number of agent-house pairs for which the indicator function was set to 1 as a consequence of strategyproofness. At the start, Algorithm 2 inserts the canonical profile R_s where all agents have the same preferences into the priority queue. The algorithm then enters a loop that evaluates a profile from the priority queue in each iteration until the priority queue is empty. This setup is very similar to a best first search algorithm which in turn is similar to breath first search and depth first search. The main difference is that it is not searching for a target profile but traversing all profiles, visiting some of them multiple times. In each iteration, the profile with the highest priority is deleted from the priority queue and evaluated by Algorithm 1. For all agent-house pairs that are determined to be equal to *RSD*, the indicator function is updated.

Then, the algorithm uses strategyproofness, or localizedness to be precise, to propagate the new information to neighboring profiles. If $f(R,i,h) = RSD(R,i,h)$ holds and agent i manipulates by rearranging houses above and below h, then $f(R',i,h) = RSD(R',i,h)$ (where R' is the profile agent i manipulates to) has to hold by strategyproofness. Therefore, we can set $I_{RSD}(R',i,h) = 1$ if $I_{RSD}(R,i,h) = 1$ since *RSD* itself satisfies strategyproofness. 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. Let R' be the profile to which agent i manipulates by swapping the houses ranked at position k and $k+1$ in $\succ_i$. Since Algorithm 2 is restricted to $\mathcal{R}^*$, it could be that $R' \notin \mathcal{R}^*$ or that there are multiple mappings from R' to its canonical form. To take care of these two problems, Algorithm 2 computes the canonical profile $R^* = \textit{canonical}(R')$ such that $R' \in [R^*]$. Furthermore, a variable *manipulators* is created that contains the set of agents that the manipulating agent i can be mapped to. Then, for each house that was not swapped and for each agent that agent i could be mapped to, $I_{RSD}(R^*,i,h) \leftarrow 1$ if $I_{RSD}(R,i,h) = 1$ and $I_{RSD}(R^*,i,h) = 0$. This step implements the transfer of information in I_{RSD} to neighboring profiles by using localizedness. A counter Δ keeps track of how many new entries profile R^* receives. At the end R^* is inserted into the priority queue with priority Δ if $\Delta > 0$, and if it is already present, its priority is increased by Δ instead.

Example 2. Let us explain an example run of the first step of the algorithm for arbitrary n. The algorithm will initialize the priority queue with the profile R_s in which all agents share the same preferences. In the first iteration of the loop, this profile is chosen from the queue as it is the only element. The profile is passed to the *QP* which evaluates it and shows that all entries must be equal to *RSD*. This follows directly from equal treatment of equals and the assignment constraints. Therefore, all entries $I_{RSD}(R)$ are set to 1. Next, the algorithm considers all possible manipulations, let the manipulating agent be $i = 1$ and let $k = 1$ be the swap index, then R' is the profile where agent 1 ranks $h_2 \succ_1 h_1 \succ \dots$ and all others rank $h_1 \succ h_2 \succ \dots$ since agent 1 swapped the houses at position 1 and 2. Now, R' is mapped to the canonical profile R^*, which looks as follows. Agent n ranks $h_2 \succ_n h_1 \succ_n \dots$ and all other agents rank $h_1 \succ h_2 \succ \dots$. The algorithm computes the set of agents that agent 1 can be mapped to from R' to R^*. In this example, agent 1 is mapped to agent n. Then, for all $l \in [n] \setminus \{k, k+1\}$, the house at position l is determined, in particular, agent 1

Algorithm 2 Verify *RSD* Characterization

Input

n Number of Agents and Objects

$I_{RSD} \leftarrow 0$ ▷ Initialize $I_{RSD} : \mathcal{R}^* \times N \times U \to \{1, 0\}$ as the constant 0 function.
$queue \leftarrow$ **new** *Priority Queue*
$queue.insert(R_s, 0)$
while $queue$ **is not** $empty$ **do**
  $R \leftarrow queue.findmax()$
  $queue.deletemax()$
  $P \leftarrow QP(R, I_{RSD})$ ▷ Determine additional entries that are *RSD* with *QP*.
  for all $(i, h) \in P$ **do**
    $I_{RSD}(R, i, h) \leftarrow 1$
  end for
  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**
    $R^*, manipulators = canonical(R')$
    $\Delta \leftarrow 0$
    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).

in R ranks h_l at the same position as agent n in R^*. Now, the condition $I_{RSD}(R, 1, h_l) = 1$ and $I_{RSD}(R^*, n, h_l) = 0$ is checked for all l. We know this will always evaluate to true since $I_{RSD}(R)$ is 1 and $I_{RSD}(R^*)$ was initialized as 0 and has not changed, yet. Therefore, we set $I_{RSD}(R^*, n, h_l)$ to 1. Since there are $n - 2$ houses for which this value is set to 1, the variable Δ that tracks the priority increase of the profile is set to $n - 2$. Next, the profile is inserted into the priority queue with priority $n - 2$. This process is then repeated for all agents and all possible swap manipulations. In this particular step, the manipulation of each agent that repeats the same manipulation as agent 1 maps to the same canonical profile. These additional manipulations do not change new entries of the indicator function to 1 and therefore don't increase the priority of the profile. Then, the algorithm finishes the iteration and since the priority queue is not empty, it will pick one of the profiles from the queue next.

To conclude this section, we now prove that Algorithm 2 is able to show that the *RSD* characterization holds in case this is true. It cannot show the other direction since the algorithm does not use nonperverseness.

Theorem 1. *If Algorithm 2 returns true for a fixed n, then RSD is the only assignment rule for n agents and houses that satisfies equal treatment of equals,* ex post *efficiency, and strategyproofness.*

Proof. We prove the statement by showing that $I_{RSD}(R, i, h) = 1 \implies f(R, i, h) = RSD(R, i, h)$ is an invariant that holds at every point of the algorithm. This is sufficient since the algorithm returns true if and only if $I_{RSD} \equiv 1$ at the end. Furthermore, Lemma 3 shows that it suffices to restrict the algorithm to the canonical domain $\mathcal{R}^*$. At the start of Algorithm 2, $I_{RSD} \equiv 0$ and the implication is trivially true for all profiles R, agents i and houses h. We show that the invariant is true in each iteration of the algorithm. In line 7, the current profile R and I_{RSD} are given as input to Algorithm 1. Lemma 1 shows that for the returned agent-house pairs (i, h), the equation $f(R, i, h) = RSD(R, i, h)$ holds. Therefore, the algorithm can set $I_{RSD}(R, i, h) = 1$ for all returned pairs (i, h). We conclude the implication is true for all triples which contain profile R. Next, the algorithm considers profile R^* that is the canonical profile of profile R' that can be reached from R by a adjacent swap manipulation of agent i in line 11. Let k be the index of the house that agent i swaps with the house at index $k + 1$, let h^* be a house that agent i does not rank at position k or $k + 1$ and let i^* be one of the agents that agent i can be mapped to. If $I_{RSD}(R, i, h) = 1$ and $I_{RSD}(R^*, i^*, h^*) = 0$ then Algorithm 2 sets $I_{RSD}(R^*, i^*, h^*) \leftarrow 1$. The first condition $I_{RSD}(R, i, h) = 1$ is sufficient for the correctness of this step since by assumption, $I_{RSD}(R, i, h) = 1$ implies $f(R, i, h) = RSD(R, i, h)$ and by strategyproofness of f and RSD, we get $f(R^*, i^*, h^*) = f(R, i, h) = RSD(R, i, h) = RSD(R^*, i^*, h^*)$ which shows $f(R^*, i^*, h^*) = RSD(R^*, i^*, h^*)$. The second condition $I_{RSD}(R^*, i^*, h^*) = 0$ ensures that the priority change Δ is only increased by one if the implication has been shown for a new triple. We conclude that the invariant holds during and after each iteration of the main while loop. Finally, if $I_{RSD}(R, i, h) \equiv 1$ then $f(R, i, h) = RSD(R, i, h)$ for all profiles R, agents i and houses h and therefore, $f = RSD$. □

4 Results

The current state of *RSD* characterizations via equal treatment of equals, *ex post* efficiency, and strategyproofness for small n are summarized in Figure 2. The first characterization for $n = 3$ was shown by Bogomolnaia and Moulin [4]. In their proof, they use a lemma that is essentially a weakening of support efficiency. Since *ex post* efficiency and support efficiency

	Extra Condition	Strategyproofness	Source
$n \leq 3$	—	strategyproofness	Bogomolnaia and Moulin [4]
$n \leq 4$	symmetry	only localizedness	Sandomirskiy [18]
$n \leq 5$	—	only localizedness	this paper

Figure 2: 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.

are equivalent for $n = 3$, the later also implies their lemma. This shows that full *ex post* efficiency is not required for $n = 3$.

Recently, Sandomirskiy [18] has shown with a computer-aided proof that the characterization holds for $n \leq 4$ using symmetry, support efficiency, and localizedness. We extend these results 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 [7]. This domain is rich enough for the impossibility of equal treatment of equals, strategyproofness, and ordinal efficiency by Bogomolnaia and Moulin [4]. 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 2. *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.

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] S. Bade. Random serial dictatorship - the one and only. *Mathematics of Operations Research*, 45(1):353–368, 2020.

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

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

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

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

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

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

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

[19] 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.

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

[21] 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 2. *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 matching. 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 $n = 3$. □

Next, we show the proofs of Lemmas 1 and 2.

Lemma 1. *Let R be a profile, E a set of agent-house pairs, and I_{RSD} the current state of the indicator function during program execution, i.e., $I_{RSD}(R,i,h)=1$ implies $f(R,i,h)=RSD(R,i,h)$. Then, for all $(i,h)\in N\times H\setminus E$, $f(R,i,h)=RSD(R,i,h)$ when the optimal objective value of the QP in Figure 1 is 0.*

Proof. The QP in Figure 1 finds an assignment for profile R that satisfies equal treatment of equals, *ex post* efficiency and has maximal $L2$-distance to $RSD(R)$. Assume $p_{i,h}$ for all $i\in N, h\in H$ and λ_π for all $\pi\in\Pi$ are a optimal solution with objective value 0. Assume now for contradiction that a pair $(i^*,h^*)\in N\times H\setminus E$ exists such that $f(R,i^*,h^*)=RSD(R,i^*,h^*)$ does not hold for all functions f that satisfy our axioms and the additional constraints imposed by I_{RSD}. Let g be such a function. Set $p^*=g(R)$ and since g is *ex post* efficient, there exists a vector λ^* such that $\sum_{\pi\in\Pi}\lambda^*_\pi SD(R,\pi)=g(R)$. This solution is feasible by assumption and the objective is strictly larger than 0 because $\sum_{(i,h)\in N\times H\setminus E}(p_{i,h}-RSD(R,i,h))^2\geq(p_{i^*,h^*}-RSD(R,i^*,h^*))^2>0$. This contradicts our assumption that we chose an optimal solution for the QP. □

Lemma 2. *Let $\pi\in\Pi$ be a permutation of the agents and $\tau\in\mathcal{T}$ a permutation of the houses. If p, λ is an optimal solution for $QP(R,I_{RSD},E)$, then $p^{\pi,\tau}$, $\pi(\lambda)$ is an optimal solution for $QP(R^{\pi,\tau},I^{\pi,\tau}_{RSD},E^{\pi,\tau})$ with the same objective value.*

Proof. Let p, λ be an optimal solution for $QP(R,I_{RSD},E)$ with objective value v. Choose $\pi\in\Pi$ and $\tau\in\mathcal{T}$ arbitrarily. We show that $p^{\pi,\tau}$, $\pi(\lambda)$ is an optimal solution for $QP(R^{\pi,\tau},I^{\pi,\tau}_{RSD},E^{\pi,\tau})$. The proof consists of three parts. First, we show that it is a feasible solution, then that it has the same objective value v and finally, that it is optimal.

We start by showing that $p^{\pi,\tau}$, $\pi(\lambda)$ is a feasible solution for $QP(R^{\pi,\tau},I^{\pi,\tau}_{RSD},E^{\pi,\tau})$. Since p satisfies all equal treatment of equals constraints, $p^{\pi,\tau}$ also satisfies the equal treatment of equal constraints because $\succ_i=\succ_j$ if adn only if $\succ_{\pi(i)}=\succ_{\pi(j)}$. For the *ex post* efficiency constraints, we have $\sum_{\alpha\in\Pi}\lambda_\alpha SD(R,\alpha)=p$ if and only if $\sum_{\pi(\alpha)\in\Pi}\lambda_{\pi(\alpha)}SD(R^{\pi,\tau},\pi(\alpha))=p^{\pi,\tau}$ because the permutations just rearrange terms on both sides. The lottery constraints obviously still hold, therefore the only remaining constraints are the constraints of the form $p_{i,h}=RSD(R,i,h)$ if $I_{RSD}(R,i,h)=1$. These become $p^{\pi,\tau}_{\pi(i),\tau(h)}=RSD(R^{\pi,\tau},\pi(i),\tau(h))$ if $I_{RSD}(R^{\pi,\tau},\pi(i),\tau(h))=1$, again it is easy to see that the constraints must still hold since we permuted the condition as well as both sides of the equation accordingly.

Next, we show that the objective value is the same for both QP. Assume $(i,h)\notin E$, then, $(p_{i,h}-RSD(R,i,h))^2$ is part of the objective by definition. Furthermore, $(\pi(i),\tau(h))\notin E^{\pi,\tau}$ and thus, $\left(p^{\pi,\tau}_{\pi(i),\tau(h)}-RSD(R^{\pi,\tau},\pi(i),\tau(h))\right)^2$ is in the objective of the other QP. Since RSD satisfies symmetry and $p^{\pi,\tau}$ is equal to p up to permutation of the agents and houses by definition, we have that $(p_{i,h}-RSD(R,i,h))^2=\left(p^{\pi,\tau}_{\pi(i),\tau(h)}-RSD(R^{\pi,\tau},\pi(i),\tau(h))\right)^2$ for all pairs (i,h). We conclude that the objective values are equivalent.

Finally, we show that the solution is optimal. Assume for contradiction it is not optimal. Then, a different solution with strictly larger objective exists. Consequently, there also exists a feasible solution for $QP(R,I_{RSD},E)$ with objective value strictly greater than v which follows from the previous arguments with permutations $\pi^{-1}\Pi$ and $\tau^{-1}\in\mathcal{T}$. All in all, we have shown the full statement that if p, λ is an optimal solution for $QP(R,I_{RSD},E)$, then $p^{\pi,\tau}$, $\pi(\lambda)$ is an optimal solution for $QP(R^{\pi,\tau},I^{\pi,\tau}_{RSD},E^{\pi,\tau})$ with the same objective value. □

Finally, we state the proof of Theorem 2.

Theorem 2. *RSD is not characterized by equal treatment of equals,* ex post *efficiency and strategyproofness in the domain $\mathcal{R}^{>}$.*

					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 3: 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 $\mathcal{R}^{>}$.

Proof. The rule defined in Figure 3 satisfies all three axioms and was found using QP. It is equal to RSD on all canonical profiles except the five shown in Figure 3. Profiles in the same equivalence class receive the same random assignment permuted accordingly. □