Soc Choice Welf
DOI 10.1007/s00355-016-0978-8

ORIGINAL PAPER

# Efficient lottery design

**Onur Kesten**[1] · **Morimitsu Kurino**[2] · **Alexander S. Nesterov**[3]

Received: 18 November 2014 / Accepted: 5 June 2016


**Abstract** There has been a surge of interest in stochastic assignment mechanisms that have proven to be theoretically compelling thanks to their prominent welfare properties. Contrary to stochastic mechanisms, however, lottery mechanisms are commonly used in real life for indivisible goods allocation. To help facilitate the design of practical lottery mechanisms, we provide new tools for obtaining stochastic improvements in lotteries. As applications, we propose lottery mechanisms that improve upon the widely used random serial dictatorship mechanism and a lottery representation of its competitor, the probabilistic serial mechanism. The tools we provide here can be useful in developing welfare-enhanced new lottery mechanisms for practical applications such as school choice.

We would like to thank two anonymous reviewers and seminar participants at various universities and conferences for useful discussions. Morimitsu Kurino acknowledges the financial support from Maastricht University when he was affiliated there. All remaining errors are our own.

✉ Onur Kesten
okesten@andrew.cmu.edu

Morimitsu Kurino
kurino@sk.tsukuba.ac.jp

Alexander S. Nesterov
alexander.nesterov@wzb.eu

1 Tepper School of Business, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213, USA

2 Faculty of Engineering, Information and Systems, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki 305-8573, Japan

3 WZB, Reichpietschufer 50, 10785 Berlin, Germany

Published online: 14 June 2016

## 1 Introduction

A lottery is a common tool to establish fairness in real-life indivisible goods allocation problems such as object/task assignment, on-campus housing, kidney exchange, course allocation, and school choice. The simplest of these problems is the so-called assignment problem, where a set of distinct objects is allocated to a set of agents. A widely used real-life mechanism for such problems is the *random serial dictatorship* (RSD): a random ordering of agents is drawn from a uniform lottery, and the first agent picks her favorite object; the second agent picks her favorite object among the remaining ones; and so on. RSD satisfies many desirable properties. Ex post efficiency is an important one: after the resolution of the lottery, the resulting deterministic assignment is Pareto efficient. In a number of school districts, where schools are equipped with possibly distinct and coarse priority orders over students, popular assignment mechanisms such as Boston and Deferred Acceptance (Gale and Shapley 1962) are applied upon randomly breaking the ties in schools' priority orders. All of these mechanisms, which we henceforth refer to as *lottery mechanisms*, induce a probability distribution over deterministic assignments, i.e., a lottery over mappings of agents to objects.

Notwithstanding the prominence and popular usage of lottery mechanisms in practice,[1] there has been much recent interest in *stochastic mechanisms* that prescribe the marginal probabilities with which each agent is assigned each object. In other words, a stochastic mechanism, unlike a lottery mechanism, does not immediately output a deterministic assignment but rather outputs a (sub)stochastic assignment matrix indicating agents' marginal assignment probabilities. To implement a stochastic mechanism one often resorts to a Birkhoff-von Neumann type of decomposition that transforms the outcome of the stochastic mechanism into an equivalent lottery over deterministic assignments. An important advantage and a chief motivation of the stochastic approach is that it makes it possible to achieve superior efficiency properties relative to lottery mechanisms. A well-known example of this approach is the *probabilistic serial* (PS) mechanism by Bogomolnaia and Moulin (2001) (hereafter BM),[2] which has become the cornerstone of a rapidly growing body of literature concerning stochastic mechanisms (cf. Che and Kojima 2010; Kojima and Manea 2010; Hashimoto et al. 2014).

BM have pointed out that the RSD outcome may suffer from unambiguous efficiency losses regardless of the von Neumann-Morgenstern utilities compatible with agents' ordinal preferences. Manea (2009) shows that these losses are prevalent even in large assignment problems. BM introduce a stronger notion of efficiency, which we call "sd-efficiency": a stochastic assignment is *sd-efficient* if it is not dominated by another stochastic assignment. Surprisingly, RSD may not always induce sd-efficient outcomes. BM have proposed PS as a serious contender to RSD, which selects the central point within the sd-efficient set. The attractive sd-efficiency (as well as the sd-

[1] Indeed we are not aware of any stochastic mechanisms in use for any practical assignment problem.

[2] PS treats each object as a continuum of probability shares and allows agents to simultaneously "eat away" from their favorite objects at the same speed until each agent has eaten a total of 1 probability share. The share of an object an agent has eaten during the process represents the probability with which she assigned the object by PS. See Sect. 5 for a more precise description.

envy-freeness) property has triggered much interest to further extend and generalize PS to richer and more structured assignment problems (cf. Kojima 2009; Athanassoglou and Sethuraman 2011; Budish et al. 2013).

An obvious advantage of lottery mechanisms is that they largely facilitate ex post analysis, which may focus on considerations such as incentives, fairness, stability, individual rationality, and efficiency. Nevertheless, the lottery approach has not been as successful as the stochastic approach as far as achieving stronger welfare properties than ex post efficiency.[3] Nevertheless, because a stochastic assignment needs to be decomposed into a feasible lottery before actual implementation (Birkhoff 1946; von Neumann 1953; Kojima and Manea 2010), ex post considerations are comparably more difficult, if not impossible, to handle in the domain of stochastic assignments.[4] Therefore, we believe that bridging the gap between the two approaches and developing tools that would allow one to work directly with lotteries without sacrificing efficiency is an important task. In this paper, our goal is to show that ex ante efficiency analysis in addition to ex post analysis can be performed directly using lotteries.

We set off on our quest by uncovering the link between ex post efficiency and sd-efficiency. In a related paper, Abdulkadiroğlu and Sönmez (2003a) study whether the sd-inefficiency of a stochastic assignment could be attributed to the Pareto inefficiency of a deterministic assignment it may induce and give a negative answer to this question.[5] We provide a complementary result to this observation. In particular, we show that for any given stochastic assignment $P$ of any given assignment problem $\succ$, there exists a corresponding deterministic assignment $\mu(P, \succ)$ that is Pareto efficient if and only if $P$ is sd-efficient at $\succ$ (Theorem 1). The deterministic assignment $\mu(P, \succ)$ is obtained by transforming the $n-$agent stochastic assignment problem into an at most $n^2-$agent deterministic assignment problem that introduces multiple replicas of each agent. An immediate corollary is Abdulkadiroğlu and Sönmez's characterization of sd-efficiency via notions of domination across sets of assignments.

An important contribution of our study, in line with the commonly used methodology and trends in indivisible goods allocation literature,[6] is to develop methods for the construction of a lottery that improves upon a given inefficient lottery while maintaining the feasibility of the final outcome (Theorem 2).[7] We observe, however, that the former part of such an objective may turn out to be quite subtle, as an ex ante welfare improvement over an ex-post lottery can actually give rise to an ex-post inefficient lottery (Example 1). For the latter part of the objective, we propose an

[3] For example, as far as we are aware, a nontrivial lottery mechanism satisfying sd-efficiency (or the stronger ex-ante efficiency) is yet to be reported or studied. Additionally imposing strategy-proofness readily leads to impossibilities (Zhou 1990; Bogomolnaia and Moulin 2001).

[4] Budish et al. (2013) develop tools for handling complex constraints while working directly with stochastic mechanisms.

[5] See Example 3 of Abdulkadiroğlu and Sönmez (2003a).

[6] Improving upon a "status quo" allocation (or a partial allocation) while respecting other considerations has been a common goal in various applications of indivisible goods allocation. Examples of applications include housing markets, on-campus housing, kidney exchange, and school choice. All these applications, however, have focused on achieving ex post properties.

[7] In a related paper, Manea (2008) shows the existence of lotteries that improve upon the RSD outcome. Differently than here, his approach is based on working directly with stochastic assignments.

algorithm that generates a feasible lottery from an infeasible lottery provided that it has a feasible equivalent. As an application of our tools and ideas, we propose new lottery mechanisms that stochastically improve upon RSD. Our proposals combine the above-mentioned methods with the celebrated object assignment method called the *top trading cycles (TTC)* method, attributed to David Gale. One of these proposals, which we call the TTC-based RSD (TRSD) mechanism, is sd-efficient, stochastically dominates RSD, and satisfies equal treatment of equals (Theorem 3).

Finally, we offer a lottery representation of PS for any given problem. The idea is based on the identification of a set of priority orders such that the equal-weight lottery over the serial dictatorship outcomes induced by the collection of these priority orders results in exactly the same stochastic assignment as the PS outcome. Recall that RSD is an equal-weight lottery over all possible priority orders of agents regardless of agents' preferences. Unlike the RSD lottery, however, the set of priority orders in the support of the lottery representation of PS, is constructed based on agents' preferences. This implies that to implement PS as a lottery mechanism, we need to elicit agents' preferences a priori and determine the set of priority orders to be used in the lottery draw. Once the support of the lottery is constructed, the rest of the assignment process proceeds in exactly the same way as with RSD: the first agent picks her favorite object; the second agent picks her favorite object among the remaining agents; and so on. We generalize this approach by proposing a lottery representation algorithm that, for any given stochastic assignment, generates an equivalent equal-weight lottery (Theorem 4).

The lottery representation algorithm enables one to transform a given stochastic mechanism into an equivalent lottery mechanism. Therefore it may be useful to supplement a stochastic mechanism with this algorithm in practical applications. Our approach in finding lottery representations to stochastic mechanisms, however, hints at a trade-off between the stochastic and the lottery approaches. Whereas the lottery construction is more transparent under the former approach (recall that the lottery approach relies on the construction of priority orders), the implementation of the final assignments is more transparent under the latter approach.

The rest of the paper is organized as follows. Section 2 introduces the model. Section 3 establishes a link between ex post and ex ante efficiency and describes our algorithm for generating a feasible lottery. Section 4 introduces the TTC-based RSD mechanisms and Sect. 5 the lottery representation of PS. Section 6 concludes.

## 2 The model

A discrete resource allocation problem is a list $(N, A, q, \succ)$ where $N = \{1, \ldots, n\}$ is a finite set of agents; $A$ is a finite set of objects; and $q := (q_a)_{a \in A}$ is a positive integer vector where $q_a$ denotes the **quota** of object $a \in A$. We assume that $|N| \leq \sum_{a \in A} q_a$; $\succ = (\succ_i)_{i \in N}$ is a preference profile where $\succ_i$ is the strict preference relation of agent $i \in N$ on $A$. Let $\succeq_i$ denote the weak relation associated with $\succ_i$. The null object, if assumed to exist, is an object in $A$ denoted by $a_0$, which is assigned a quota of $n$ so that all agents can simultaneously consume it. Agents who are assigned the null object are viewed as taking their outside options. We fix $N$, $A$, and $q$ throughout the paper, and denote a problem by a preference profile $\succ$.

A **(deterministic) assignment** is a function $\mu : N \to A$. Moreover, it is **feasible** if for each $a \in A$, $|\mu^{-1}(a)| \leq q_a$. Let $\mathcal{D}$ be the set of all assignments, and let $\mathcal{D}^f$ be the set of all feasible assignments. A feasible assignment $\mu$ is **Pareto efficient** at $\succ$ if there is no $\mu' \in \mathcal{D}^f$ such that for all $i \in N$, $\mu'(i) \succeq_i \mu(i)$, and for some $i \in N$, $\mu'(i) \succ_i \mu(i)$. A **deterministic mechanism** associates a feasible assignment with each problem.

A **stochastic allotment** is a probability distribution $P_i := (p_{i,a})_{a \in A}$ over $A$ where $p_{i,a}$ denotes the probability that agent $i$ receives object $a$, and thus for each $a \in A$, $0 \leq p_{i,a} \leq 1$ and $\sum_{b \in A} p_{i,b} = 1$. A **stochastic assignment** $P = [P_i]_{i \in N} = [p_{i,a}]_{i \in N, a \in A}$ is a substochastic matrix such that for each $i \in N$ and each $a \in A$, $\sum_{b \in A} p_{i,b} = 1$ and $\sum_{j \in N} p_{j,a} \leq q_a$. Let $\mathcal{S}$ be the set of all stochastic assignments. A **stochastic mechanism** associates a stochastic assignment with each problem.

**Definition 1** A **lottery** $L = \sum_{s \in S} w_s \mu_s$ is a probability distribution over assignments such that

- (L1) set $S$, called an **index set**, is nonempty and finite;
- (L2) $\sum_{s \in S} w_s = 1$;
- (L3) for each $s \in S$, $0 < w_s \leq 1$ and $w_s$ is a rational number; and
- (L4) for each $s \in S$, $\mu_s \in \mathcal{D}$,

where $w_s$ is called the **weight** of $\mu_s$, and $\mu_S = (\mu_s)_{s \in S} \in \mathcal{D}^S$ is the **support** of $L$. Moreover, it has **equal weights** if for each $s \in S$, $w_s = 1/|S|$ and it is **feasible,** if instead of (L4), it satisfies (L4'): for each $s \in S$, $\mu_s \in \mathcal{D}^f$.

Note that the support is a product set, contrary to the standard terms.[8] Also note that the index set is finite and the weights are rational numbers.[9] A **(feasible) lottery mechanism** associates a (feasible) lottery with each problem.

For each assignment $\mu \in \mathcal{D}$, let $\pi(\mu)$ be a $|N| \times |A|$ matrix that represents $\mu$. Note that a given feasible lottery $L = \sum_s w_s \mu_s$ induces the stochastic assignment $P = \sum_s w_s \pi(\mu_s)$. Therefore, every feasible lottery mechanism can be uniquely represented as a stochastic mechanism. Given any stochastic assignment, the well-known Birkhoff-von Neumann theorem states that there is at least one feasible lottery that induces it. However, a stochastic mechanism may not be uniquely represented as a feasible lottery mechanism.

We say that two lotteries are **equivalent** if they induce the same stochastic assignment. The following is a useful lemma.

**Lemma 1** *For each lottery, there is an equivalent equal-weight lottery.*

This result follows from duplicating assignments and expanding the original index set. See the Appendix for the proof.

---

[8] The reason for this will be clear when relating the sd-efficiency of a lottery with the Pareto efficiency of an assignment in a replica economy in the next section.

[9] This tractability assumption holds generally in practice and is satisfied by lotteries induced by all well-known mechanisms.

### 2.1 The random serial dictatorship mechanism (RSD)

We introduce a popular lottery mechanism, called the random serial dictatorship, which will be our focus in this paper. To this end we use a **priority** of agents in $N$ that is a bijection from $\{1, 2, \ldots, |N|\}$ to $N$. For example, given a priority $f$, $f(1)$ is the agent with the highest priority, $f(2)$ is the one with the second-highest priority, and so on. Let $F$ be the set of all priorities.

Next is the **serial dictatorship (deterministic) mechanism induced by a priority** $f \in F$. We denote it by $SD_f$. Fix a problem $\succ$. The assignment $SD_f(\succ)$ is found iteratively as follows.

*Step 1:* The highest priority agent $f(1)$ is assigned her top-choice object under $\succ_{f(1)}$.

$\vdots$

*Step* $\underline{k}$: The $k$th highest priority agent $f(k)$ is assigned her top-choice object under $\succ_{f(k)}$ among the remaining objects.

Now we are ready to define the **random serial dictatorship mechanism (RSD)**, denoted by $RSD$: Fix a problem $\succ$. First, a priority $f$ is chosen with probability $1/n!$. Second, agents are assigned objects according to $SD_f(\succ)$. Formally,

$$RSD(\succ) = \frac{1}{n!} \sum_{f \in F} SD_f(\succ).$$

Note that RSD is a lottery mechanism and its index set is the set $F$ of all priorities.

### 2.2 The probabilistic serial mechanism (PS)

For each problem $\succ$, the stochastic assignment of the **probabilistic serial mechanism (PS)** is computed via the following simultaneous eating algorithm:[10] Given a problem $\succ$, think of each object $a$ as an infinitely divisible good with supply $q_a$ that agents eat in the time interval [0, 1].

*Step 1:* Each agent eats away from her top-choice object at the same unit speed. Proceed to the next step when some object is completely exhausted.

$\vdots$

*Step* $\underline{k}$: Each agent eats away from her top-choice object from her remaining ones at the same unit speed. Proceed to the next step when some object is completely exhausted.

The algorithm terminates after some step when each agent has eaten exactly 1 total unit of objects (i.e., at time 1). The stochastic allotment of an agent $i$ by PS is then given by the amount of each object she has eaten until the algorithm terminates. Let $PS(\succ)$ be the stochastic assignment of PS for problem $\succ$.

[10] See Hugh-Jones et al. (2014) for an experimental evaluation of PS.

### 2.3 Axioms

A feasible lottery is **ex-post efficient** if it can be represented as a probability distribution over Pareto-efficient feasible assignments. BM propose an appealing ex ante notion of sd-efficiency that also implies ex post efficiency, which we introduce next. Fix a problem $\succ$. Given $i \in N$ and $P, R \in \mathcal{S}$, $P_i$ **stochastically dominates** $R_i$ at $\succ_i$ if for each $a \in A$, $\sum_{b \in A: b \succeq_i a} p_{i,b} \geq \sum_{b \in A: b \succeq_i a} r_{i,b}$. In addition, $P$ **weakly stochastically dominates** $R$ at $\succ$ if for each $i \in N$, $P_i$ stochastically dominates $R_i$ at $\succ_i$. $P$ **stochastically dominates** $R$ at $\succ$ if $P$ weakly stochastically dominates $R$ at $\succ$ and $P \neq R$. A stochastic assignment is **sd-efficient** at $\succ$ if it is not stochastically dominated by another stochastic assignment at $\succ$. Next is a much weaker efficiency property. A stochastic assignment $P \in \mathcal{S}$ is **non-wasteful** at $\succ$ if for each $i \in N$, each $a \in A$ with $p_{i,a} > 0$, and each $b \in A$ with $b \succ_i a$, we have $\sum_{j \in N} p_{j,b} = q_b$. Sd-efficiency implies ex post efficiency and non-wastefulness, but not vice versa.

We define our fairness axiom. A stochastic assignment $P \in \mathcal{S}$ satisfies the **equal treatment of equals** at $\succ$ if for all $i, j \in N$, $\succ_i = \succ_j$ implies $P_i = P_j$.

Axioms of a lottery mechanism except ex post efficiency are defined for its induced stochastic assignment for each preference profile. A stochastic (lottery) mechanism is said to satisfy a property if for each preference profile, its (induced) stochastic assignment satisfies that property.

A stochastic mechanism $\varphi$ is **sd-strategy-proof** if for each problem $\succ$, each $i \in N$, and each preference $\succ_i$, $\varphi_i(\succ)$ stochastically dominates $\varphi_i(\succ_i', \succ_{-i})$ at $\succ_i$. A lottery mechanism is sd-strategy-proof if its induced stochastic mechanism is sd-strategy-proof.

A **stochastic mechanism** $\varphi$ **weakly stochastically dominates a stochastic mechanism** $\psi$ if for each problem $\succ$, $\varphi(\succ)$ weakly stochastically dominates $\psi(\succ)$. Moreover, a **stochastic mechanism** $\varphi$ **stochastically dominates a stochastic mechanism** $\psi$ if $\varphi$ weakly stochastically dominates $\psi$ and for some problem $\succ$, $\varphi(\succ)$ stochastically dominates $\psi(\succ)$ at $\succ$. Similarly, we can define the stochastic dominance of a lottery mechanism by looking at its induced stochastic mechanism.

*Remark 1* RSD is known to be sd-strategy-proof, ex-post efficient, and to satisfy the equal treatment of equals. However, it is wasteful (Erdil 2014) and thus is not sd-efficient (Bogomolnaia and Moulin 2001). Moreover, PS is known to be sd-efficient and to satisfy the equal treatment of equals but not be sd-strategy-proof (Bogomolnaia and Moulin 2001).

## 3 Sd-efficiency and Pareto efficiency

### 3.1 Characterization of sd-efficiency

Abdulkadiroğlu and Sönmez (2003a) investigate a possible link between sd-efficiency and Pareto efficiency. In particular, they ask whether the lack of sd-efficiency of a stochastic assignment (or equivalently, the sd-inefficiency of all lotteries it induces) can be associated with the lack of Pareto efficiency of a feasible assignment induced by it. They show that such a link between the two efficiency notions fails to exist: even if every feasible assignment in the support of every feasible lottery that induces a stochastic assignment is Pareto efficient, this may not be sufficient to guarantee the sd-efficiency of this feasible lottery. Our first objective is to recover the link between the two efficiency notions—albeit in a different sense—through an intuitive characterization result. We show that the sd-efficiency of a given feasible lottery is in fact implied by (and does imply) the Pareto efficiency of a "special" allocation constructed from the support of this feasible lottery. Before stating this result more precisely, we need the following definition.

**Definition 2** Let $\succ$ be a problem and $S$ be an index set. We rename $N$ as the set of types. In the **$|S|$-fold replica problem**, for each type $i \in N$, there are $|S|$ agents; for each object $a \in A$, the quota is $q_a|S|$; for each type $i \in N$, all $|S|$ agents of that type share the common preferences $\succ_i$ on $A$. Let $i_s$ be the agent of type $i$ indexed by $s \in S$, $N_s = \{1_s, \ldots, i_s, \ldots, n_s\}$ be the set of all agents indexed by $s$, and $N_S := \cup_{s \in S} N_s$ be the set of all agents. We say that $\succ_{N_s} := (\succ_{i_s})_{i_s \in N_s}$ is the **$s$-replica problem**, and $\succ_S := (\succ_{N_s})_{s \in S}$ denotes the $|S|$-fold replica problem.

An **$|S|$-fold replica assignment** is a function $\nu_S : N_S \to A$ such that for each $a \in A$, $|\nu_S^{-1}(a)| \leq q_a|S|$. Let $\mathcal{D}_S$ be the set of all $|S|-$fold replica assignments. Given $\nu_S \in \mathcal{D}_S$ and $s \in S$, an **$s$-replica assignment** is a function $\nu_s : N_s \to A$ such that for each $i_s \in N_s$, $\nu_s(i_s) = \nu_S(i_s)$. Thus we denote $\nu_S = (\nu_s)_{s \in S}$. Note that the $s$-replica assignment $\nu_s$ from an $|S|$-fold replica assignment $\nu_S$ can be thought of as an assignment for the original problem $\succ$, but need not be feasible in the original. Thus we introduce the following. An $|S|-$fold replica assignment $\nu_S = (\nu_s)_{s \in S}$ is **feasible** if for each $s \in S$, $s-$replica assignment $\nu_s$ is feasible, i.e., for each $a \in A$, $|\nu_s^{-1}(a)| \leq q_a$.

Now we relate an $|S|-$fold replica assignment with the support of a lottery. Given a support $\mu_S = (\mu_s)_{s \in S}$ of a lottery, **the $|S|$-fold replica assignment induced by the support** $\mu_S$ is the $|S|-$fold replica assignment where for all $s \in S$, each agent $i_s \in N_s$ is assigned object $\mu_s(i_s)$. Conversely, given an $|S|-$fold replica assignment $\nu_S$, the **support (of a lottery) induced by the $|S|$-fold replica assignment** $\nu_S$ is the support in which at each $s \in S$, each agent $i \in N$ is assigned object $\nu_s(i_s)$. Note that a lottery with induced support does not always induce a stochastic assignment. It does, however, if its weights are equal.

**Lemma 2** *The equal-weight lottery with the support induced by an $|S|-$fold replica assignment produces a stochastic assignment.*

The proof is omitted as it is straightforward. By Lemma 2, from now on, unless confusion arises, the support of an equal-weight lottery is an $|S|-$fold replica assignment, and vice versa.

An $|S|-$fold replica assignment $\mu_S$ **Pareto dominates** an $|S|-$fold replica assignment $\mu'_S$ at $\succ_S$ if for all $i_s \in N_S$, $\mu_S(i_s) \succeq_i \mu'_S(i_s)$ and for some $i_s \in N_S$, $\mu_S(i_s) \succ_i \mu'_S(i_s)$. Also, an $|S|-$fold replica assignment is **Pareto efficient** at $\succ_S$ if it is not Pareto dominated by any other $|S|-$fold replica assignment. The following result relates the Pareto dominance of $|S|-$fold replica assignments with the stochastic dominance of the equal-weight lottery with induced support.

**Lemma 3** *Let $S$ be an index set, and $\mu_S$, $\mu'_S$ be $|S|-$fold replica assignments. Suppose that $\mu_S$ Pareto dominates $\mu'_S$ at $\succ_S$. Then, the equal-weight lottery with support $\mu_S$ stochastically dominates the equal-weight lottery with support $\mu'_S$ at $\succ$.*

We omit the straightforward proof. The following result links the sd-efficiency of a (feasible or infeasible) lottery and the Pareto efficiency of its support in the $|S|-$fold replica problem.

**Theorem 1** *Let $\succ$ be a problem and $L$ a lottery with an index set $S$. Then, lottery $L$ is sd-efficient at $\succ$ if and only if the support of $L$ is Pareto efficient at $\succ_S$.*

The characterization of sd-efficiency given by Theorem 1 is quite intuitive. Theorem 1 also forms the basis of a practical test of sd-efficiency as it uses the standard notion of Pareto efficiency for the support of a lottery in its replica problem. Whereas determining whether a stochastic assignment is stochastically dominated or not may be difficult, checking for the Pareto efficiency of the support of a lottery is fairly straightforward by drawing on the top trading cycles (TTC) method, which we later describe.[11]

### 3.2 An alternative proof of an sd-efficiency characterization

Based on Theorem 1, we next provide an alternative proof of Abdulkadiroğlu and Sönmez's (2003a) characterization of sd-efficiency. To this end, we introduce some notion: an $|S|-$fold replica assignment $\mu_S$ is **frequency equivalent** to an $|S|-$fold replica assignment $\nu_S$ if for each $a \in A$, $|\mu_S^{-1}(a)| = |\nu_S^{-1}(a)|$. Their characterization is based on the following notion of domination. For exposition without additional notation, we adapt their notion in our replica problem.

**Definition 3** Given an index set $S$, a feasible $|S|$-fold replica assignment $\mu'_S$ **AS dominates** an $|S|$-fold replica assignment $\mu_S$ if

1. there is an $|S|$-fold replica assignment $\bar{\mu}_S$ that is fequency equivalent to $\mu'_S$, and
2. there is a one-to-one function $f : S \to S$ such that
   (a) for each $s \in S$, $\bar{\mu}_s$ Pareto dominates or is equal to $\mu_s$ at $\succ$ and
   (b) there is $s \in S$ such that $\bar{\mu}_s$ Pareto dominates $\mu_s$ at $\succ$.

**Corollary 1** (Abdulkadiroğlu and Sönmez 2003a) *Given a problem $\succ$, let feasible lottery $L := \sum_{s \in S} w_s \mu_s$ be an arbitrary decomposition of a stochastic assignment $P$. $P$ is sd-efficient at $\succ$ if and only if for each $T \subseteq S$, $\mu_T = (\mu_t)_{t \in T}$ is AS undominated.*

The proof of Corollary 1 is immediate from the following lemma and Theorem 1. Our alternative proof has the advantage of being more transparent and shorter than the original proof of Abdulkadiroğlu and Sönmez (2003a) as our argument involves only elementary application of standard notions of Pareto efficiency to replica problems.

[11] Simply apply the TTC to the problem where the support of the lottery is interpreted as an extended housing market with endowments. Then the following is easy to show. The support of the lottery is Pareto efficient if and only if the TTC algorithm generates only self-cycles.

**Lemma 4** *Let $\succ$ be a problem and S be an index set. Then $\mu_S$ is Pareto undominated if and only if for each $T \subseteq S$, $\mu_T$ is AS undominated.*

*Proof* We prove the contrapositive of each direction. ($\Leftarrow$): If $\nu_S$ Pareto dominates $\mu_S$, then it is straightforward to see that $\nu_S$ AS dominates $\mu_S$. ($\Rightarrow$): Suppose that for some $T \subseteq S$, some $\mu'_T$ AS dominates $\mu_T$. Then there is an $|T|$-fold replica assignment $\bar{\mu}_T$ that is frequency equivalent to $\mu'_T$; and there is a one-to-one function $f : T \to T$ such that (a) for each $s \in T$, $\bar{\mu}_s$ Pareto dominates or is equal to $\mu_s$ at $\succ$ and (b) there is $s \in T$ such that $\bar{\mu}_s$ Pareto dominates $\mu_s$ at $\succ$. Then $\bar{\mu}_T$ Pareto dominates $\mu_T$ in the $|T|$-fold replica problem. Define $\nu_S$ as for each $s \in S$, $\nu_s = \bar{\mu}_s$; otherwise $\nu_s = \mu_s$. Then $\nu_S$ Pareto dominates $\mu_S$ in the $|S|$-fold replica problem. □

### 3.3 Welfare improvement from an ex-post efficient lottery

In later sections, we aim to show that ex ante efficiency analysis as well as ex post analysis can be performed directly using lotteries. But before doing so, we make a useful observation about a possible ex post welfare consequence of stochastically improving upon a given feasible lottery. The next example shows that an ex ante welfare improvement over an ex-post efficient feasible lottery may actually entail an ex-post *inefficient* lottery.

*Example 1* (Ex ante welfare improvement over an ex-post efficient lottery results in an ex-post inefficient lottery) Let $N = \{1, 2, 3, 4\}$, $A = \{a, b, c, d\}$, and $q_a = q_b = q_c = q_d = 1$. Preferences are as follows.

$$\begin{array}{c|cccc} \succ_1 & a & b & c & d \\ \succ_2 & a & b & c & d \\ \succ_3 & b & a & d & c \\ \succ_4 & b & a & d & c \end{array}$$

Consider the following ex-post efficient lottery.

$$L = \frac{1}{2}\begin{pmatrix} 1 & 2 & 3 & 4 \\ a & b & d & c \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 1 & 2 & 3 & 4 \\ c & d & b & a \end{pmatrix}, \quad \text{and} \quad \pi(L) = \begin{pmatrix} 1/2 & 0 & 1/2 & 0 \\ 0 & 1/2 & 0 & 1/2 \\ 0 & 1/2 & 0 & 1/2 \\ 1/2 & 0 & 1/2 & 0 \end{pmatrix}.$$

Next consider the following feasible lottery.

$$L' = \frac{1}{2}\underbrace{\begin{pmatrix} 1 & 2 & 3 & 4 \\ a & c & b & d \end{pmatrix}}_{\mu_1} + \frac{1}{2}\underbrace{\begin{pmatrix} 1 & 2 & 3 & 4 \\ c & b & d & a \end{pmatrix}}_{\mu_2}, \quad \text{and} \quad \pi(L') = \begin{pmatrix} 1/2 & 0 & 1/2 & 0 \\ 0 & 1/2 & 1/2 & 0 \\ 0 & 1/2 & 0 & 1/2 \\ 1/2 & 0 & 0 & 1/2 \end{pmatrix}.$$

Clearly, lottery $L'$ stochastically dominates lottery $L$. However, the support of $L'$ contains the Pareto inefficient assignment $\mu_2$. Thus $L'$ is not ex-post efficient. We can show that there is no other feasible lottery that induces the stochastic assignment $\pi(L')$.

Given that sd-efficiency implies ex post efficiency, the observation in Example 1 is counterintuitive. It implies that ex post efficiency is not preserved under welfare improvements in stochastic assignments. One of our objectives in this paper is to develop a method for constructing a new feasible lottery that stochastically improves upon a given sd-inefficient feasible lottery $L$ while also ensuring ex post efficiency. To this end, we first take an equal-weight lottery with support $\mu_S$ equivalent to $L$ (Lemma 1), and then by correspondence of the support and $|S|-$fold replica assignment, we consider a Pareto improvement from $\mu_S$ in the $|S|-$fold replica problem. However, there is a complication in the approach of obtaining a stochastically improving lottery: even if the initial lottery is feasible, the resulting lottery induced by a Pareto improvement may not be feasible. Thus, in Sect. 3.4, we propose a method that transforms a given infeasible lottery into an equivalent feasible one, and then in Sect. 4.2, we introduce a method of Pareto improvement in the replica problem with endowments.

### 3.4 Feasible-assignment-generating (FAG) algorithm

Given an equal-weight but infeasible lottery with support $\mu_S = (\mu_s)_{s\in S}$, we introduce an algorithm that generates an equivalent and feasible lottery. Note that as we defined in Sect. 3.1, an $|S|-$fold replica assignment $\nu_S$ is **feasible** if for each $s \in S$ and each $a \in A$, $|\nu_s^{-1}(a)| \leq q_a$.

**Feasible assignment generating (FAG) algorithm**

**Initialization** Given is an $|S|-$fold replica assignment $\mu_S = (\mu_s)_{s\in S}$. Without loss of generality, assume $S = \{1, 2, \dots, |S|\}$. We focus on swapping objects in the set $\bar{A} := \left\{a \in A \mid |\mu_S^{-1}(a)| > 0\right\}$—those that are assigned under $\mu_s$ for some $s \in S$. For given $i \in N$ and $s \in S$, $\mu_s(i)$ is sometimes denoted by $\mu_S(s, i)$. We use both notations whenever convenient. Let $\mu_S(S, i) = \left\{\mu_S(s, i) \in \bar{A} \mid s \in S\right\}$ and $\mu_S(1, I) = \left\{\mu_S(1, i) \in \bar{A} \mid i \in I\right\}$. Given $O \subseteq \bar{A}$, let

$$B(O) = \cup_{i\in N:\, \mu(1,i)\in O} \{\mu_S(S, i)\},$$

$$B^t(O) = \begin{cases} O & \text{if } t = 1, \\ B(B^{t-1}(O)) & \text{if } t \geq 2. \end{cases}$$

**Phase 1 (Swap path identification)** Let $a \in \mu_S(1, |S|)$ such that $|\mu_1^{-1}(a)| > q_a$, i.e., object $a$ is assigned more agents than its quota at $\mu_1$ (if no such object exists, $\mu_1$ is feasible and we are done.). Let $X = \left\{c \in \bar{A} \mid |\mu_1^{-1}(c)| \leq q_c - 1\right\}$, i.e., the set of objects that are only partially assigned to agents at $\mu_1$ under $\mu_S$. Check if $B^1(\{a\}) \cap X \neq \emptyset$; if not, check if $B^2(\{a\}) \cap X \neq \emptyset$; ...; and so on. Let $t \in \mathbb{N}$ be the smallest number such that $B^t(\{a\}) \cap X \neq \emptyset$. This procedure is well defined by the following claim (see the Appendix for the proof).

*Claim 1* (1) $B^0(\{a\}) \subseteq B^1(\{a\}) \subseteq B^2(\{a\}) \subseteq \dots$;
(2) For each $t \in \{0\} \cup \mathbb{N}$, if $B^t(\{a\}) \cap X = \emptyset$, then $B^t(\{a\}) \subsetneq B^{t+1}(\{a\})$;

(3) There is $t \in \{0\} \cup \mathbb{N}$ such that $B^t(\{a\}) \cap X \neq \emptyset$. Thus, $\{a\} \subsetneq B^1(\{a\}) \subsetneq \cdots \subsetneq B^t(\{a\})$.

**Phase 2 (Execution of swaps)** Phase 1 implies that there are $(t+1)$, $t \geq 1$, distinct objects $b_0 := a, b_1, \ldots, b_t := x$ such that $b_1 \in B(\{b_0\})$, $b_2 \in B(\{b_1\})$, ..., $b_t = x \in B(\{b_{t-1}\}) \cap X$. This implies that there are $t$ distinct agents, $i_1, i_2, \ldots, i_t$, and corresponding indices, $k_{i_1}, k_{i_2}, \ldots, k_{i_t}$ such that $\mu_S(1, i_1) = b_0 = a$ and $\mu_S(k_{i_1}, i_1) = b_1$; $\mu_S(1, i_2) = b_1$ and $\mu_S(k_{i_2}, i_2) = b_2$; ...; $\mu_S(1, i_t) = b_{t-1}$ and $\mu_S(k_{i_t}, i_t) = b_t = x$. Next update the support $\mu_S$ by setting $\mu_S(1, i_1) := b_1$ and $\mu_S(k_{i_1}, i_1) := b_0 = a$; $\mu_S(1, i_2) := b_2$ and $\mu_S(k_{i_1}, i_2) := b_1$, ..., $\mu_S(1, i_t) := b_t$ and $\mu_S(k_{i_t}, i_t) := b_{t-1}$.

**Iteration** Given the support $\mu_S$, repeating Phases 1 and 2 at most $n-1$ times yields a new support $\mu_S^1$ whose first index assignment, $\mu_1^1$, is feasible. Thus, we have finalized the first index assignment. Next we obtain a new support $\mu_S^2$, whose first index assignment coincides with that of $\mu_S^1$, by iteratively applying Phases 1 and 2 to the subsupport obtained from $\mu_S^1$ by restricting to the assignments from 2 to $|S|$. Thus we have finalized the second index assignment. Continuing similarly the algorithm terminates once we have cleared indices 1 through $|S|-1$. The final support $\mu_S^{|S|-1}$ consists of $|S|$ feasible assignments. Therefore, we obtain the following.

**Proposition 1** *Given an $|S|-$fold replica assignment $\mu_S$, the FAG algorithm produces a feasible $|S|-$fold replica assignment that is frequency equivalent to $\mu_S$.*

The following is a corollary of Lemma 1 and Proposition 1.

**Corollary 2** *Given any infeasible lottery, there is an equivalent feasible lottery with equal weights.*

We call a stochastic assignment **rational** if all of its entries are rational numbers. Then we can straightforwardly represent a rational stochastic assignment by an equal-weight infeasible lottery. Thus, as a corollary of Proposition 1, we have

**Corollary 3** *Any rational stochastic assignment can be expressed as a feasible equal-weight lottery that induces it.*

*Remark 2* Note that Corollary 3 gives a version of Birkhoff (1946), von Neumann (1953) when the stochastic assignment is restricted to be rational. An advantage of FAG, which we believe merits further investigation, could be that in order to generate a feasible assignment it allows the designer to swap individual assignments of agents. This could provide more flexibility in choosing which final deterministic assignments should emerge and in so doing may help meet other ex post constraints the designer may have in mind (e.g., diversity and complementarities). This direction is left for future work.

*Example 2* (Finding feasible assignment) Let $N = \{1, 2, 3, 4, 5, 6\}$ and $A = \{a, b, c, d, e, f\}$ such that all of the objects have the quota of 1. Consider the following support $\mu_S = (\mu_1, \mu_2, \mu_3)$.

$$\mu_1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ a & a & b & c & d & e \end{pmatrix}, \quad \mu_2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ a & c & c & d & f & e \end{pmatrix}, \quad \text{and} \quad \mu_3 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ b & b & d & e & f & f \end{pmatrix}.$$

**Initialization.** We first tabulate these assignments into a table:

$$\mu_S = \begin{pmatrix} \mathbf{a} & [a] & \mathbf{b} & [c] & \mathbf{d} & [e] \\ a & [c] & c & d & \mathbf{f} & e \\ \mathbf{b} & b & \mathbf{d} & [e] & f & [f] \end{pmatrix}.$$

**Phase 1 (Swap path identification)** Observe that object $a$ is assigned to multiple agents at $\overline{\mu}_1$ although $q_a = 1$; and object $f$ is not assigned to any agent at $\mu_1$, i.e., $X = \{f\}$. We start with $B(\{a\}) = \{a, b, c\}$. Since $B(\{a\}) \cap X = \emptyset$, we proceed with $B^2(\{a\}) = B(\{a, b, c\}) = \{a, b, c, d, e\}$. Since $B^2(\{a\}) \cap X = \emptyset$, we proceed with $B^3(\{a\}) = B(\{a, b, c, d, e\}) = A \setminus \{a_0\}$. Since $B^3(\{a\}) \cap X = \{f\}$, we conclude that $t = 3$.
**Phase 2 (Execution of swaps)** From Phase 1 we easily obtain a set of four objects $\{b_0 = a, b_1 = b, b_2 = d, b_3 = f\}$ such that $b \in B(\{a\})$, $d \in B(\{b\})$, and $f \in B(\{d\})$. In particular, we obtain a corresponding set of three agents $\{1, 3, 5\}$ such that $\mu(1, 1) = a$ and $\mu(3, 1) = b$; $\mu(1, 3) = b$ and $\mu(3, 3) = d$; and $\mu(1, 5) = d$ and $\mu(2, 5) = f$. The agents and their assignments identified in this fashion are indicated in boldface in the above table. (Note that such agent and object sets may not be uniquely obtained. An alternative path from object $a$ to $f$ is indicated in brackets in the above table.) Next we execute the vertical swaps to update the table as follows:

$$\mu_S = \begin{pmatrix} \mathbf{b} & \mathbf{a} & \mathbf{d} & \mathbf{c} & \mathbf{f} & \mathbf{e} \\ a & c & c & d & d & e \\ a & b & b & e & f & f \end{pmatrix}.$$

**Iteration** Observe that the first row of the updated table above induces a feasible assignment, which is indicated in boldface. So we next reapply Phases 1 and 2 to the remaining two rows. Then it is not very difficult to see that the remaining table contains two trivial vertical swaps involving agent 5 and either of agents 2 and 3 for swapping object $c$ with $b$, and object $d$ with $f$. The following is one possible final table whose three rows induce the feasible assignments $\mu_1$, $\mu_2$, and $\mu_3$ respectively.

$$\mu_S = \begin{pmatrix} b & a & d & c & f & e \\ a & b & c & d & f & e \\ a & c & b & e & d & f \end{pmatrix}.$$

## 4 Lottery mechanisms dominating the random serial dictatorship mechanism

The most widely used lottery mechanism in real-life markets is the random serial dictatorship mechanism (RSD). However, as BM pointed out, RSD is not sd-efficient but only ex-post efficient. In this section, we propose a method of improving upon RSD.

### 4.1 Efficient lottery construction (ELC) procedure

We shall propose a method, the **efficient lottery construction (ELC)** procedure, to directly construct an sd-efficient lottery that stochastically dominates a given equal-weight sd-inefficient lottery $L = \frac{1}{|S|}\sum_{s\in S}\mu_s$. For a given problem $\succ$, our procedure is as follows.

*Stage 1* (Improvement). We consider the $|S|$-fold replica problem $\succ_S$ with endowments $\mu_S = (\mu_s)_{s\in S}$ where each agent $i_s \in N_S$ owns an object $\mu_s(i_s)$. Note that because $L$ is sd-inefficient, by Theorem 1, its support $\mu_S$ is Pareto inefficient in the replica problem. Then we apply a Pareto improvement algorithm (to be introduced in the next subsection), which selects a Pareto efficient assignment $\nu_S$.

*Stage 2* (FAG algorithm). We apply the FAG algorithm (Sect. 3.4) to obtain a feasible $|S|-$fold replica assignment $\nu_S^f$.

*Stage 3* (New lottery). Take the equal-weight lottery $L' := \frac{1}{|S|}\sum_{s\in S}\nu_s^f$.

**Theorem 2** *For each problem $\succ$ and each feasible sd-inefficient lottery $L$, the ELC algorithm induces an sd-efficient lottery that stochastically dominates $L$.*

*Proof* Because $\nu_S$ is Pareto efficient at the replica problem, by Theorem 1, the induced lottery is sd-efficient. Moreover, $\nu_S$ Pareto dominates $\mu_S$, by Lemma 3, and lottery $L$ dominates lottery $L'$. □

### 4.2 Top trading cycles (TTC) algorithm

We introduce a Pareto-improving algorithm that we alluded to in the ELC procedure. This is based on the well-known idea of Gale's top trading cycles (Shapley and Scarf 1974). The top trading cycles (TTC) algorithm was originally introduced for a housing market where each object is owned by only one agent.[12] In contrast, we deal with replica problems with endowments where an object is owned by multiple agents. For this reason, we introduce a priority $g \in F$ as if an object were owned by only the highest-priority owner.

For a given priority $g \in F$ and a given replica problem $\succ_S$ with endowments $\mu_S$, the **TTC algorithm** induces an $|S|-$fold replica assignment as follows:

*Step 0* For each object $a \in A$, assign a counter that keeps track of how many copies of the object are available. Initially set the counter equal to $q_a|S|$.

*Step 1* Each agent $i_s \in N_S$ points to her favorite object according to $\succ_i$. Each object points to the highest-priority type among those who own the object according to priority $g$. If there are several agents of the same type, pick one of them arbitrar-

[12] Because of its appealing efficiency and incentive features, a number of mechanisms based on the TTC method have been proposed and characterized for a variety of applications such as on-campus housing, school choice, and kidney exchange. Although for deterministic settings, all proposed TTC based mechanisms are Pareto efficient, little is known about the applicability of this procedure to the stochastic assignment context or its relation to sd-efficiency, for that matter. An exception is Kesten (2009) who shows that if a simple version of the TTC method is applied to a market in which each agent is initially endowed with an equal probability share of each object, then the resulting outcome is sd-efficient and coincides with that of PS.

ily. There is at least one cycle where a cycle is a finite list of objects and agents $(a^1, i^1, a^2, i^2, \ldots, a^m, i^m)$ such that each agent $i^\ell$ points to object $a^\ell$ $(\ell \in \{1, \ldots, m\})$, and agent $i^m$ points to object $a^1$. Each agent in a cycle is assigned a copy of the object that she is pointing to and is removed. The counter of each object in the cycle is reduced by one, and if it reduces to zero, the object is also removed. Counters of all the other objects stay the same.

*Step k* Each remaining agent $i_s$ points to her favorite object among the remaining ones according to $\succ_i$. Each remaining object points to the highest-priority remaining type according to priority $g$. If there are several agents of the same type, pick one of them arbitrarily. There is at least one cycle. Each agent in a cycle is assigned a copy of the object that she is pointing to and is removed. The counter of each object in the cycle is reduced by one, and if it reduces to zero, the object is also removed. Counters of all the other objects stay the same.

The above algorithm terminates in a finite step when all agents are assigned objects.

*Last step* Note that the assignment $\nu_S$ induced by the above algorithm is not always feasible in the sense that some $s$-replica assignment $\nu_s$ is not feasible in the original problem $\succ$. For this reason, we apply the FAG algorithm to obtain a feasible assignment, which we denote by $TTC_S(\succ_S, \mu_S, g)$.

Note that the TTC algorithm implements Stages 1 and 2 in the ELC procedure. Now we are ready to state Proposition 2 (the proof is omitted, as the idea is very similar to the one for the Shapley and Scarf's (1974) for the housing market):

**Proposition 2** *For each $|S|-$fold replica problem $\succ$ with endowments $\mu_S$, the TTC algorithm induces a Pareto efficient assignment at $\succ_S$ that Pareto dominates $\mu_S$ and is equal to $\mu_S$ when $\mu_S$ is Pareto efficient at $\succ_S$.*

### 4.3 TTC-based random serial dictatorship$^K$ mechanism (TRSD$^K$)

Using the ELC procedure and the TTC algorithm discussed in Sects. 4.1 and 4.2, we propose an ex-post efficient lottery mechanism that dominates RSD and satisfies the equal treatment of equals—what we call the TTC-based random serial dictatorship$^K$ mechanism (TRSD$^K$) given a natural number $K \in \{1, \ldots, n!\}$.

Let us consider how to improve upon RSD. With our tools developed so far—in particular—Lemma 3, we need to convert the problem into a replica problem with endowments. Ideally it is best to take the set of priorities, $F$, as the index set for the replica problem. However, as the number of agents, $n$, becomes large, the size of $F$, $n!$, becomes huge and computationally difficult to work on. To avoid this problem, we pick only $|K|$ distinct priorities, $f_1, \ldots, f_K$, and then consider the improvement over the induced random serial dictatorship $\frac{1}{K}\sum_{k=1}^{K} SD_{f_k}(\succ)$ by using the improving method of the TTC discussed in the previous subsection. This is the key idea of our TRSD$^K$ mechanism, which we introduce next.

Let a problem $\succ$ and $K \in \{1, \ldots, n!\}$ be given.

*Step 1* We choose $K$ distinct priorities $f_1, \ldots, f_K$ out of all $n!$ priorities with equal probability $1/\binom{n!}{K}$ where the set $F$ of all priorities have $n!$ priorities, and $\binom{n!}{K}$ is the number of $K-$combinations from $n!$ elements. Let $F(K) := \{f_1, \ldots, f_K\}$.

*Step 2* We consider an improvement of the lottery $\frac{1}{K}\sum_{k=1}^{K} SD_{f_k}(\succ)$ that is a lottery of choosing SD assignments $SD_{f_k}(\succ)$ with priority $f_k$ being selected with equal priority $1/K$. Moreover, we choose a priority $g \in \{f_1, \ldots, f_K\}$ with equal probability $1/K$. Then we apply the TTC algorithm for the priority $g$ to the problem $\succ_S$ with endowments $(SD_{f_k}(\succ))_{k=1}^{K}$, and then we obtain the $|K|$-fold replica assignment $TTC_{F(K)}(\succ_{F(K)}, SD_{F(K)}(\succ); g)$. Then we consider the induced equal-weight lottery $\frac{1}{K}\sum_{f\in F(K)} TTC_f(\succ_{F(K)}, SD_{F(K)}(\succ); g)$.

We denote the resulting lottery by $TRSD^K(\succ)$, and can express it as

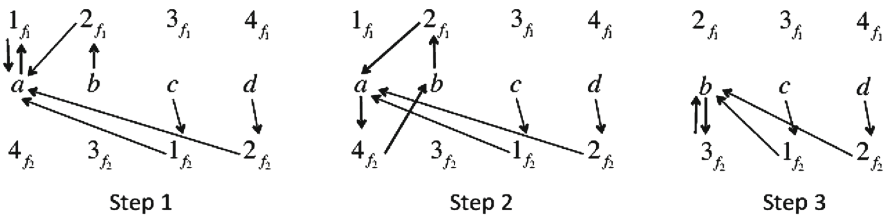
$$TRSD^K(\succ) = \frac{1}{|\mathcal{F}(K)|}\sum_{F(K)\in\mathcal{F}(K)} \frac{1}{K}\sum_{g\in F(K)} \frac{1}{K}\sum_{f\in F(K)} TTC_f(\succ_{F(K)}, SD_{F(K)}(\succ); g), \quad (1)$$

where $\mathcal{F}(K) := \{\{f_1, \ldots, f_K\} \mid f_1, \ldots, f_K \in F \text{ are distinct}\}$ and $|\mathcal{F}(K)| = \binom{n!}{K}$. Note that TRSD$^1$ coincides with RSD.

*Example 3* We show how to implement $TRSD^K$ where $K \geq 2$. Consider $K = 2$ and an example where $N = \{1, 2, 3, 4\}$, $A = \{a, b, c, d\}$, $q_a = q_b = q_c = q_d = 1$. Let preferences be given by:

| $\succ_1$ | $\succ_2$ | $\succ_3$ | $\succ_4$ |
|---|---|---|---|
| $a$ | $a$ | $b$ | $b$ |
| $b$ | $b$ | $a$ | $a$ |
| $c$ | $c$ | $c$ | $c$ |
| $d$ | $d$ | $d$ | $d$ |

Suppose that $f_1 = (1, 2, 3, 4)$ and $f_2 = (3, 4, 1, 2)$ are chosen and $g = f_1$. Then $SD_{f_1}(\succ) = \begin{pmatrix} 1 & 2 & 3 & 4 \\ a & b & c & d \end{pmatrix}$ and $SD_{f_2}(\succ) = \begin{pmatrix} 1 & 2 & 3 & 4 \\ c & d & b & a \end{pmatrix}$. Then we apply the TTC algorithm as follows:




Here, for simplicity, we draw only the pointing arrows from agents who are also pointed at by objects, and we skip the remaining steps. We obtain $\begin{pmatrix} 1 & 2 & 3 & 4 \\ a & a & c & d \end{pmatrix}$ and $\begin{pmatrix} 1 & 2 & 3 & 4 \\ c & d & b & b \end{pmatrix}$. Then, applying the FAG algorithm, we obtain $TTC_{f_1}(\succ_{F(K)}, SD_{F(K)}(\succ); f_1) = \begin{pmatrix} 1 & 2 & 3 & 4 \\ a & d & c & b \end{pmatrix}$ and $TTC_{f_2}(\succ_{F(K)}, SD_{F(K)}(\succ); f_1) = \begin{pmatrix} 1 & 2 & 3 & 4 \\ c & a & b & d \end{pmatrix}$. One of these two assignments is selected with 1/2 as a result of $TRSD^2$.

**Theorem 3** *Let $K \in \{2, \ldots, n!\}$. TRSD$^K$ is ex-post efficient, weakly stochastically dominates $RSD$, and satisfies the equal treatment of equals. Moreover, we have the following.*[13]

1. *Suppose the unit quotas of all objects, i.e., for each $a \in A$, $q_a = 1$. If $|N| \leq 3$ then $TRSD^K = RSD$. If $|N| \geq 4$, then $TRSD^K$ stochastically dominates $RSD$.*
2. *If $RSD$ is not sd-efficient for some $N$, $A$, and $q$, then there is $\bar{K} \leq |A|$ such that for each $K \geq \bar{K}$, $TRSD^K$ stochastically dominates $RSD$.*
3. *For some $N$, $A$, $q$, and $K$, $TRSD^K$ is not sd-efficient.*

Clearly, TRSD$^K$ is more tractable and practical when $K$ is small. Note that $K$ can be as large as $|F| = n!$. Part (2) asserts that in the standard model (Bogomolnaia and Moulin 2001), we can improve upon RSD by taking only $K = 2$. Moreover, Part (3) asserts that in general $K$ can be a small number relative to $n!$ in order to improve upon RSD.

*Proof* We first show the ex post efficiency. By Proposition 2, $TTC_{F(K)}(\succ_{F(K)}, SD_{F(K)}(\succ); g)$ is Pareto efficient in the $K$-fold replica problem. Thus $TTC_f(\succ_{F(K)}, SD_{F(K)}(\succ); g)$ is Pareto efficient at the original problem $\succ$. Hence $TRSD^K$ is ex-post efficient.

We next show that $TRSD^K$ weakly stochastically dominates $RSD$. $RSD$ can be expressed as

$$RSD(\succ) = \frac{1}{n!} \sum_{f \in F} SD_f(\succ) = \frac{1}{\binom{n!}{K}} \sum_{F(K) \in \mathcal{F}(K)} \frac{1}{K} \sum_{g \in F(K)} \frac{1}{K} \sum_{f \in F(K)} SD_f(\succ), \quad (2)$$

for each problem $\succ$. We compare (1) with (2): for each $F(K) \in \mathcal{F}(K)$ and each $g \in F(K)$, by Proposition 2, $TTC_{F(K)}(\succ_{F(K)}, SD_{F(K)}; g)$ Pareto dominates or coincides with $SD_{F(K)}(\succ)$. Thus, by Lemma 3, TRSD$^K$ weakly stochastically dominates RSD.

In the Appendix we prove the equal treatment of equals and the stochastic dominance in Parts (1) and (2). It remains to show Part (3) – the sd-inefficiency. Let $N = \{1, 2, 3, 4\}$, $A = \{a, b, a_0\}$, $q_a = q_b = 1$, and $\succ$ such that for each $i \in \{1, 2\}$, $a \succ_i b \succ_i a_0$; and for each $i \in \{3, 4\}$, $b \succ_i a \succ_i a_0$. Here $a_0$ is the null object. The computational simulation gives us the following stochastic assignments.

Then we can see the following assignment $P$ stochastically dominates $TRSD^2$ and $TRSD^3$: for each $i \in \{1, 2\}$, $P_i = (0.5, 0, 0.5)$; for each $i \in \{3, 4\}$, $P_i = (0, 0.5, 0.5)$. □

[13] It is quite challenging to check whether or not the TRSD$^K$ is sd-strategy-proof, for the following reasons. First, BM's and Nesterov's (2014) impossibility theorems show the incompatibility of sd-strategy-proofness, sd-efficiency, and equal treatment of equals for problems with unit quotas. Thus their results are not applicable since TRSD$^K$ is not necessarily sd-efficient in general, and nor does our setting assume unit quotas. Second, we need at least four agents for the outcomes of RSD and TRSD$^K$ to differ, which makes it cumbersome to calculate the stochastic assignments of TRSD$^K$.

| | $RSD$ | | | $TRSD^2$ | | | $TRSD^3$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | $a$ | $b$ | $a_0$ | $a$ | $b$ | $a_0$ | $a$ | $b$ | $a_0$ |
| Agent 1 | 0.4167 | 0.0833 | 0.5000 | 0.4312 | 0.0688 | 0.5000 | 0.4417 | 0.0583 | 0.5000 |
| Agent 2 | 0.4167 | 0.0833 | 0.5000 | 0.4312 | 0.0688 | 0.5000 | 0.4417 | 0.0583 | 0.5000 |
| Agent 3 | 0.0833 | 0.4167 | 0.5000 | 0.0688 | 0.4312 | 0.5000 | 0.0583 | 0.4417 | 0.5000 |
| Agent 4 | 0.0833 | 0.4167 | 0.5000 | 0.0688 | 0.4312 | 0.5000 | 0.0583 | 0.4417 | 0.5000 |

## 5 Lottery representation of the probabilistic serial mechanism

Motivated by the sd-inefficiency of RSD, BM introduced a central stochastic mechanism that achieves sd-efficiency—the probabilistic serial mechanism (PS). However, since PS is not a lottery mechanism, it might be less tempting to implement in practice, as discussed in the Introduction. In this section, we offer an algorithm of representing a PS stochastic assignment by an equal-weight lottery. Specifically, for each problem $\succ$, we construct a collection of priorities $F^* := (f_j)_{j=1}^J$, such that

$$PS(\succ) = \frac{1}{|J|}\sum_{j=1}^{J} SD_{f_j}(\succ).$$

Note that

$$RSD(\succ) = \frac{1}{|F|}\sum_{f\in F} SD_f(\succ),$$

where $F$ is the set of all priorities. The differences between the set $F$ and the collection $F^*$ are threefold: (i) $F^*$ depends on preference profiles, (ii) $F^*$ might contain fewer different priorities than $F$ does, and (iii) $F^*$ will usually contain several copies of some of the priorities.[14] Before we proceed to the algorithm, consider the following motivating example.

*Example 4* Let $N = \{1, 2, 3, 4\}$, $A = \{a, b, c, d\}$, and $q_a = q_b = q_c = q_d = 1$. Consider the following problem and its PS assignment

$$\begin{array}{c|cccc} \succ_1 & a & b & c & d \\ \succ_2 & a & b & d & c \\ \succ_3 & a & c & d & b \\ \succ_4 & a & d & c & b \end{array} \qquad PS(\succ) = \begin{pmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & 0 \\ \frac{1}{4} & \frac{1}{2} & 0 & \frac{1}{4} \\ \frac{1}{4} & 0 & \frac{3}{4} & 0 \\ \frac{1}{4} & 0 & 0 & \frac{3}{4} \end{pmatrix}.$$

where in the eating algorithm of PS, object $a$ is first exhausted at time $1/4$, then object $b$ at time $3/4$, and then objects $c, d$ last at the same time of 1.

If we try to construct the lottery and the corresponding collection $F^*$ for $PS(\succ)$, we first see that in any possible priority the first-priority agent always receives object

[14] In fact, we can show a more general result in which any sd-efficient stochastic assignment (and not only PS) can be represented as an equal-weight lottery using the same algorithm.

$a$ under SD. Thus there are at least four different priorities in $F^*$, and each of them begins with one of the four agents. It is logical to assume that the objects exhausted earlier in the algorithm should also be assigned earlier in the lottery representation. Let us assume that in each priority the objects are assigned in the following order: $a, b, c, d$.

Consider a priority $f_1$ such that agent 1 has the first priority. Since object $b$ is split between agents 1 and 2, only agent 2 can have the second priority. Similarly, agents 3 and 4 follow and thus $f_1 = (1, 2, 3, 4)$. The same logic applies if we begin another priority $f_2$ with agent 2. Then the only feasible sequence is $f_2 = (2, 1, 3, 4)$.

However, if we begin the ordering with agent 3, it is not clear whether agent 1 or agent 2 should receive the second priority, since they both have positive probability for object $b$. But if agent 1 follows and gets object $b$, then in the next step there is no agent left to be assigned object $c$, since neither agent 2 nor agent 4 receive it in expectation. Therefore, the only feasible ordering starting with agent 3 is $f_3 = (3, 2, 1, 4)$. Similarly, $f_4 = (4, 1, 3, 2)$.

In total, we have only four feasible priorities. Moreover, since object $a$ is split equally between all agents, the weights of these priorities are equal to $\frac{1}{4}$. Therefore $F^* = (f_j)_{j=1}^4$.

We now use the intuition from Example 4 to construct a general algorithm in the following steps: First order the objects. Then determine the set of feasible priorities (in the example there were only four). Next find the corresponding maximum weights of these priorities in the resulting lottery.[15] Finally calculate the individual contribution of each specific priority in the equal-weight lottery.

We begin by ordering the objects. Consider a problem $\succ$ and a stochastic assignment $P = PS(\succ)$. Let us relabel the objects as $a_1, a_2, \ldots, a_k$, in the exhausting order (denoted as $l_{ex}$) in the eating algorithm of PS. When two or more objects are simultaneously exhausted we order them arbitrarily. The objects that have only been partially exhausted are put in the end of the ordering in some arbitrary order: $a_{k+1}, \ldots, a_{|A|}$. For each object $a_j$, let $E(a_j)$ be the set of agents who have eaten $a_j$: $E(a_j) = \{i \in N | p_{i,a_j} > 0\}$.[16]

Next, given $P$ and $l_{ex}$, for each priority $f \in F$, we determine the maximum weight $m(f, P, l_{ex}) := min_{j<|N|} p_{f(j),a_j}$—the minimum of the assignment probabilities in $P$ that correspond to agents in $f$ and objects in $l_{ex}$ (where $f(j)$ denotes the agent that has $j^{th}$ priority in $f$). We refer to this weight as the maximum, since the final weight of this priority $f$ in the lottery will be no higher than $m(f, P, l_{ex})$. If, for instance, the first agent in $f$ does not belong to $E(a_1)$ and thus $p_{f(1),a_1} = 0$, then the overall maximum weight of $f$ is zero and $f$ does not enter the final lottery.

15 This step is missing in the example since all priorities have the same weight.

16 For a general case of an arbitrary sd-efficient assignment, objects can also be relabeled according to the exhausting order, although the underlying eating algorithm proceeds not using constant eating speed functions but some other profile of eating speed functions (Bogomolnaia and Moulin 2001). Alternatively, we can order the objects using the following hierarchical procedure: at each step agents point to their most preferred object among the remaining objects and we choose the most popular object (choose one of them arbitrarily if there are several) to be the next in our order of objects. Intuitively, this ordering of objects is similar to the exhausting order in the eating algorithm: at each step $j$ the agents in $E(a_j)$ prefer object $a_j$ over the remaining objects. This is the key feature of the ordering $l_{ex}$ in our lottery decomposition.

Importantly, for each $f$ such that $m(f, P, l_{ex}) > 0$ the order in which the objects are picked in a serial dictatorship $SD_f$ coincides with $l_{ex}$. This property follows from two facts. First, if $m(f, P, l_{ex}) > 0$, then each agent $f(j)$ has a positive probability share $p_{f(j),a_j} > 0$ of object $a_j$. For example, first-ordered agent $f(1)$ has a positive probability share $p_{f(1),a_1} > 0$ of object $a_1$, second-ordered agent $f(2)$ has a positive probability share $p_{f(2),a_2} > 0$ of object $a_2$, and so on. Second, ordering $l_{ex}$ gives the following hierarchy for objects: in the PS eating algorithm a higher object is exhausted (weakly) earlier. Thus, each agent with a positive share of $a_1$ top-ranks $a_1$ (otherwise $a_1$ cannot be exhausted first); each agent with a positive share of $a_2$ either top-ranks $a_2$ or, alternatively, top-ranks $a_1$ and ranks $a_2$ as second, and so on. Therefore, in a serial dictatorship agent $f(j)$ picks $a_j$, and the order in which the objects are picked coincides with $l_{ex}$ (the same argument holds for different definitions of $l_{ex}$ discussed in footnote 16).

Among all priorities $f \in F$ we (arbitrarily) pick one of the priorities with the *lowest positive* maximum weight and denote it as $f_1$: $f_1 \in \arg\min_f \{m(f, P, l_{ex}) \mid m(f, P, l_{ex}) > 0\}$. This priority $f_1$ enters the resulting lottery with weight $m_1 = m(f, P, l_{ex})$.

Having determined $f_1$ and $m_1$, we subtract the corresponding assignment from the old assignment matrix (denoted as $P_1 = P$). This way we get the updated matrix $P_2 := P_1 - m_1 SD_{f_1}(\succ)$. We then repeat the previous two stages for this updated matrix $P_2$ and continue doing so until all the relevant priorities together with the corresponding weights are determined. Then, similar to Lemma 1, we turn the lottery into the equal-weight lottery. Meanwhile, the set of relevant priorities that we picked at each stage becomes the collection of priorities $F^*$ that defines the equal-weight lottery.

We now formally define the algorithm for the special case when the number of agents is the same as the total amount of objects ($|N| = \sum_{a \in A} q_a$), which implies that the objects have the unit quotas (for each $a \in A$, $q_a = 1$), and $P$ is a bistochastic matrix.

**Definition 4** (*Lottery representation algorithm*) Given $|N| = \sum_{a \in A} q_a$ and $(\succ, P, l_{ex})$, the lottery representation algorithm constructs the collection of priorities $F^*$ as follows.
*Stage 1* Let $P_1 = P$. Calculate $m(f, P_1, l_{ex}) := min_{j<|N|} p_{f(j),a_j}$ for each priority $f \in F$. Among all the priorities, pick $f_1 \in \arg\min_{f \in F} \{m(f, P_1, l_{ex}) \mid m(f, P_1, l_{ex}) > 0\}$ – one with the lowest positive maximum weight, denote the corresponding weight as $m_1 = m(f_1, P_1, l_{ex})$.
*Stage j* Update the probability matrix as $P_j := P_{j-1} - m_{j-1} SD_{f_{j-1}}(\succ)$. For matrix $P_j$, find the priority with the lowest maximum weight, $f_j \in \arg\min_{f \in F} \{m(f, P_j, l_{ex}) \mid m(f, P_j, l_{ex}) > 0\}$. Denote $m_j = m(f_j, P_j, l_{ex})$.
*Final stage r*$\underline{r}$ The updated matrix becomes null, i.e., $P_r = P_{r-1} - m_{r-1} SD_{f_{r-1}}(\succ) = 0$.
Given $\{f_j, m_j\}_{j=1}^r$, we construct the required collection $F^*$ by finding the least common multiple for all the inverted weights $\frac{1}{m_0}$ and including each of the corresponding priority $f_j$ in $F^*$ precisely $\frac{m_j}{m_0}$ times.

The following theorem shows that the proposed iterative procedure is always feasible and results in the equal-weight lottery equivalent to the initial stochastic assignment $P$.

**Theorem 4** *Given $|N| = \sum_{a\in A} q_a$, for each problem $\succ$ and a stochastic assignment $P$ containing only rational elements and which is sd-efficient at $\succ$, the lottery representation algorithm induces an equal-weight lottery that is equivalent to $P$.*

*Proof* We first show the feasibility of operations at all stages of the algorithm. We show (a) the existence of the lowest maximum weight and the corresponding priority at any stage of the procedure and (b) the feasibility of updating the stochastic assignment matrix, given that we found the lowest maximum weight at the preceding stage. Then we make sure that (c) the algorithm terminates and that the representation is correct.

(a) We first show by induction that at any stage $j < r$, the matrix $P_j$ is quasi-bistochastic (its columns and rows sum up to the same positive number). The claim is correct for $P_1 = P$. Assume it also holds for $P_{j-1}$. Due to the Birkhoff-von Neumann theorem, $P_{j-1}$ can be decomposed as a convex combination of assignments (note that the lowest weight in this convex combination is weakly lower than the lowest element in $P_{j-1}$). Each assignment $\mu$ corresponds to some priority $f$ defined as follows: the agent matched with $a_1$ receives the first priority in $f$, the agent matched with $a_2$ receives the second priority in $f$, and so on along $l_{ex}$. All such priorities $f$ have positive maximum weights $m(f, P_{j-1}, l_{ex})$, which we define as the minimum element in $P_{j-1}$ among the elements that correspond to assignment $SD_f(\succ)$. Among those priorities we pick $f_{j-1}$ – the priority with the lowest positive maximum weight $m_{j-1}$.

(b) Given $f_{j-1}$ and $m_{j-1}$, we update the assignment matrix as $P_j = P_{j-1} - m_{j-1}SD_{f_{j-1}}(\succ)$. In doing so, we subtract a positive number that was smaller than the lowest positive element in $P_{j-1}$ from precisely one element in each row and in each column of $P_{j-1}$; we do not subtract from zero elements (otherwise $f_{j-1}$ is not feasible). Thus $P_j$ remains quasi-bistochastic.

(c) At each stage of the algorithm, the updated stochastic assignment contains at least one more zero element. Therefore, the algorithm terminates in $r \leq (|N|^2 - |N|)$ stages, since at the last stage $r$ the stochastic assignment matrix degenerates into a weighted assignment $SD_{f_r}(\succ)$. It is straightforward from the updating formula to check whether $\sum_{j=1}^{r} m_j SD_{f_j}(\succ) = P$.

□

Now we extend the algorithm to the case when there are fewer agents than objects: $|N| \leq \sum_{a\in A} q_a$. We use a simple trick: for each problem $\succ$ and each stochastic assignment $P$, we add the total of $(\sum_{a\in A} q_a - |N|)$ artificial agents. The preferences $\succ'$ of each artificial agent $i'$ are such that he prefers the objects that were originally left in expectation, i.e., with the total assignment probabilities being less than one, to the objects that were consumed fully: $a_l \succ'_{i'} a_j$, where $j < k+1 \leq l$. The preferences of the normal agents remain as before in $\succ$. Since the total number of agents is now the same as the number of objects, the assignment probabilities for the artificial agents are such that the modified stochastic assignment matrix $P'$ becomes bistochastic.

We then run the lottery representation algorithm for the triple $(\succ', P', l_{ex})$, where the preferences and the stochastic assignment matrix include the artificial agents, but the order of objects $l_{ex}$ remains the same as for $(\succ, P)$. After we receive the collection of priorities $F'^{*}$, we take out all the artificial agents from each of the priorities. The agents that were below the artificial agents in some priority $f'$ now get a higher slot.

It is easy to see that the result of the new lottery is precisely $P$. First, in $P'$ the artificial agents consumed only those probability shares that were not taken by normal agents in $P$; given Theorem 4, the same holds for the lottery representation of $P'$. However, when we take some artificial agent $i'$ out of some priority $f'$, given the preferences of the artificial agents, each normal agent $i$ that was below $i'$ in $f'$ receives the same object that he received before agent $i'$ was taken out. Therefore, if we take out all the artificial agents in $f'$, then the assignment of normal agents does not change, and neither does the weight of $f'$ in the lottery.

## 6 Concluding remarks

In this paper, we have introduced new tools that allow the designer to work directly with lotteries and enhance the efficiency properties of existing lottery mechanisms. Whereas the stochastic approach has already proved extremely useful in achieving superior welfare features over its lottery counterparts, coupling lottery-type assignment methods with the tools developed here may help close the gap between the two approaches while also benefiting from the practical appeal of lottery mechanisms.

Our analysis of the construction of ex post and sd-efficient lotteries lends itself to new interpretations of the workings of the prominent mechanisms RSD and PS. Abdulkadiroğlu and Sönmez (1998) show that the lottery produced by RSD is equivalent to a lottery constructed in the following way: Start from the initial lottery that assigns an equal probability (namely, $\frac{1}{n!}$) to each feasible assignment, and apply the TTC algorithm to each feasible assignment in the support of the initial lottery and replace feasible assignment by the corresponding outcome of the algorithm. Since the TTC algorithm produces Pareto efficient feasible assignments, such a lottery is ex-post efficient but not sd-efficient (as is the one induced by RSD). Kesten (2009) shows that the stochastic assignment produced by PS is equivalent to a stochastic assignment constructed in the following way: Start from an initial stochastic assignment that endows each agent each object with the same probability (namely, $\frac{1}{n}$) and apply the TTC algorithm (that considers self and pairwise-cycles) in a way that allows each agent to trade assignment probabilities of her most-preferred object with *every* other agent who is endowed with a positive probability for this object. Our analysis indicates that the difference between RSD and PS derives from the way they choose the improvement cycles from among those induced by the support of the initial lottery. Whereas RSD considers only those top trading cycles induced by each feasible assignment in the support of the initial lottery individually, PS considers all the top trading cycles induced by all feasible assignments in the support of the initial lottery altogether.

In the United States, many school districts use centralized clearinghouses to determine student assignments to public schools (Abdulkadiroğlu and Sönmez 2003b). In school choice, each school has multiple capacity and is assigned a priority order of students by the school district to be used while determining student assignments. In many school districts student priorities are typically coarse, giving rise to weak priority orders. As a consequence, school districts rely on lottery mechanisms that use randomization to generate strict priority orders by breaking the ties among equal-priority students via lottery draws. Although an assignment problem is a special school choice problem with each school having unit capacity and all students having equal priority for all schools, our analysis can be generalized straightforwardly and adapted to school choice problems, and in particular, could be helpful in improving the ex ante efficiency of school choice lotteries (see Kesten and Ünver 2015).

## Appendix

*Proof of Lemma 1* Let $L = \sum_{s \in S} w_s \mu_s$ be a lottery. The lemma is obvious if $S$ is a singleton. Thus, suppose not. Without loss of generality, let $S = \{1, \ldots, |S|\}$. By (L2) and (L3), for some $n \in \mathbb{N}$, for each $s \in S$, there is $m_s \in \mathbb{N}$ such that $w_s = m_s/n$ and $\sum_{s \in S} m_s = n$. Then, $\pi(L) = \pi\left(\sum_{s \in S} \frac{m_s}{n}\mu_s\right) = \pi[\frac{1}{n}\sum_{s \in S}(\overbrace{\mu_s + \ldots + \mu_s}^{m_s})]$. We iteratively define a collection of sets, $\{M_s\}_{s \in S}$: $M_1 = \{1, \ldots, m_1\}$, for $s \geq 2$, $M_s = \left\{\sum_{k=1}^{s-1} m_k + 1, \ldots, \sum_{k=1}^{s-1} m_k + m_s\right\}$. Moreover, let $M = \cup_{s \in S} M_s$. Also, we define a collection of assignments, $(\nu_m)_{m \in M}$ as follows: for each $m \in M$, since there is a unique $s \in S$ with $m \in M_s$, let $\nu_m = \mu_s$. Then, the lottery $\frac{1}{n}\sum_{m \in M} \nu_m$ is of equal weights and equivalent to $L$. □

*Proof of Claim 1* Part (1) is obvious by construction of $B^t(\cdot)$.
Part (2) Let $t \in \{0\} \cup \mathbb{N}$. Suppose $B^t(\{a\}) \cap X = \emptyset$, but $B^t(\{a\}) = B^{t+1}(\{a\})$. Let $\{i_1, \ldots, i_M\} := \left\{i \in I \mid \mu_S(1, i) \in B^t(\{a\})\right\}$, and for each $m \in \{1, \ldots, M\}$, let $a_m := \mu(1, i_m) \in B^t\{a\}$. Since $B^t(\{a\}) \cap X = \emptyset$, $a_m \notin X$, i.e., for each $m \in \{1, \ldots, M\}$, $|\mu_1^{-1}(a_m)| \geq q_{a_m}$. This inequality is strict for at least one $m$, as $\{a\} \in B^t(\{a\})$ and $|\mu_1^{-1}(a)| > q_a$. Thus, $\sum_{a \in \{a_1, \ldots, a_m\}} |\mu_1^{-1}(a)| = \sum_{m=1}^{M} |\mu_1^{-1}(a_m)| > \sum_{m=1}^{M} q_{a_m} \geq \sum_{a \in \{a_1, \ldots, a_m\}} q_a$, which contradicts the feasibility of $\mu_S$.
Part (3) If the claim is not true, we have $\{a\} \subsetneq B^1(\{a\}) \subsetneq \cdots \subsetneq B^t(\{a\}) \subsetneq \ldots$, which contradicts the finiteness of $A$. □

To prove Theorems 1 and Part (2) of Theorem 3, we need the following notion and lemma.

**Definition 5** Let $\succ \in \mathbf{P}^N$ and $P, R \in \mathcal{S}$. A **temporary list of size *m*** is $(a^1, i^1, \ldots, a^m, i^m, a^{m+1})$ such that for each $\ell \in \{1, \ldots, m\}$, (1) $a^{\ell+1} \succ_{i^\ell} a^l$, (2) $p_{i^\ell, a^\ell} < r_{i^\ell, a^\ell}$, (3) $p_{i^\ell, a^{\ell+1}} > r_{i^\ell, a^{\ell+1}}$, and (4) $a^1, \ldots, a^m$ are distinct. An **improvement cycle from *R* to *P*** is a temporary list of size $m$, $(a^1, i^1, \ldots, a^m, i^m, a^{m+1})$ , such that $a^{m+1} = a^1$.

**Lemma 5** *Let $\succ \in \mathbf{P}^N$, $i \in N$, and $P, R \in \mathcal{S}$ be non-wasteful at $\succ$. Suppose that $P$ stochastically dominates $R$ at $\succ$. Then there is an improvement cycle from $R$ to $P$.*

*Proof of Claim 1* We first construct a temporary list of size 1, $(a^1, i^1, a^2)$, where $a^1$ and $a^2$ are distinct. Since $P \neq R$, there is $i^1 \in N$ such that $P_{i^1} \neq R_{i^1}$. Thus, since $P_{i^1}$ stochastically dominates $R_{i^1}$ at $\succ_{i^1}$, there are $a^1, a^2 \in A$ such that $a^2 \succ_{i^1} a^1$, $p_{i^1,a^2} > r_{i^1,a^2}$, and $p_{i^1,a^1} < r_{i^1,a^1}$. Thus $a_1 \neq a_2$. Then $(a^1, i^1, a^2)$ is the desired list.

Suppose we are given a temporary list of size $m$, $(a^1, i^1, \ldots, a^{m-1}, i^{m-1}, a^m)$, where $a^1, \ldots, a^m$ are distinct. Then (1) $a^m \succ_{i^{m-1}} a^{m-1}$, (2) $p_{i^{m-1},a^{m-1}} < r_{i^{m-1},a^{m-1}}$, and (3) $p_{i^{m-1},a^m} > r_{i^{m-1},a^m}$. Then, since $r_{i^{m-1},a^{m-1}} > p_{i^{m-1},a^{m-1}} \geq 0$, by the feasibility of $P$ and non-wastefulness of $R$, we have $\sum_{j\in N} p_{j,a^m} \leq q_{a^m} = \sum_{j\in N} r_{j,a^m}$. Thus, since $p_{i^{m-1},a^m} > r_{i^{m-1},a^m}$, there is $i^m \in N$ such that $p_{i^m,a^m} < r_{i^m,a^m}$. Thus, since $P_{i^m}$ stochastically dominates $R_{i^m}$ at $\succ_{i^m}$, there is $a^{m+1} \in A$ such that $a^{m+1} \succ_{i^m} a^m$ and $p_{i^m,a^{m+1}} > r_{i^m,a^{m+1}}$. Thus $(a^1, i^1, \ldots, a^m, i^m, a^{m+1})$ is a temporary size of $m$. Then, if $a^{m+1} = a^\ell$ for some $\ell \in \{1, \ldots, m\}$, then the list $(a^\ell, i^\ell, \ldots, a^m, i^m, a^{m+1})$ is an improvement cycle from $R$ to $P$. Otherwise we continue this process. However, since $|A|$ is finite, we eventually obtain an improvement cycle from $R$ to $P$. □

*Proof of Theorem 1* Let $L$ be a lottery with the support $\mu_S$: ($\Rightarrow$) We show the contrapositive. Suppose that the support $\mu_S$ of $L$ is not Pareto efficient at $\succ_S$. Then there is an $|S|-$fold replica assignment $\nu_S$ that Pareto dominates $\mu_S$ at $\succ_S$. As in Lemma 1, there is an equal-weight lottery $L^e = (1/|M|)\sum_{m\in M} \mu'_m$ that is equivalent to $L$ such that for each $m \in M$ there is a unique $s(m) \in S$ with $\mu'_m = \mu_{s(m)}$. Now we define an $|S|-$fold replica assignment $\nu'_M$: for $m \in M$, $\nu'_m = \nu_{s(m)}$. Then, $\nu'_M$ Pareto dominates $\mu'_M$ at $\succ_M$. By Lemma 3, the equal-weight lottery with the support $\nu'_M$ stochastically dominates the equal-weight lottery $\mu'_M$ at $\succ$. Thus $L$ is not sd-efficient at $\succ$.

($\Leftarrow$) We show the contrapositive. Suppose that $L$ is wasteful (and thus not sd-efficient) at $\succ$. Let $R = \pi(L)$ be the stochastic assignment induced by $L$. Then there is $i \in N$, $a \in A$ with $r_{i,a} > 0$, and $b \in A$ with $b \succ_i a$ such that $\sum_{j\in N} r_{j,b} < q_b$. As $r_{i,a} > 0$, there is $s \in S$ such that $\mu_s(i_s) = a$. Then, let $\nu_s$ be an $s-$replica assignment such that $\nu_s(i_s) = b$ and for each $j \in N$, $\nu_s(j_s) = \mu_s(j_s)$. Then, the $|S|-$fold replica assignment $(\nu_s, \mu_{S\setminus\{s\}})$ Pareto dominates $\mu_s$ at $\succ_S$.

Suppose that $L$ is non-wasteful but not sd-efficient at $\succ$. Then, there is a stochastic assignment $P \neq R$ that stochastically dominates $R$ at $\succ$. By Lemma 5, there is an improvement cycle, denoted by $(a^1, i^1, \ldots, a^m, i^m)$, from $R$ to $P$. Then, we can find indices $s^1, \ldots, s^m \in S$ such that $\mu_{s^1}(i^1) = a^1, \ldots, \mu_{s^m}(i^m) = a^m$. Then, define an $|S|-$fold replica assignment $\nu_S$ such that $\nu_{s^1}(i^1) = a^2, \ldots, \nu_{s^{m-1}}(i^{m-1}) = a^m$, $\nu_{s^m}(i^m) = a^1$, and any other agent is assigned the same object as in $\mu$. Then, $\nu_S$ Pareto dominates $\mu_S$ at $\succ_S$. □

*Proof of Theorem 3* We first show that TRSD$^K$ satisfies the equal treatment of equals. Let $i, j \in N$ with $i \neq j$ and $\succ$ a problem with $\succ_i = \succ_j$. For each priority $f$ we define another priority $f^{i\leftrightarrow j}$ to be the priority where only the positions of $i$ and $j$ under $f$ are switched and the other agents have the same positions as in $f$. Note that the size of the support is $|\mathcal{F}(K)| \times K \times K$. Consider the lottery of the TRSD$^K$ after $F(K) = \{f_1, \ldots, f_K\} \in \mathcal{F}(K)$ is selected. Then agents face lottery $\frac{1}{K}\sum_{g\in F(K)} TTC_{F(K)}(\succ_{F(K)}, SD_{F(K)}(\succ); g)$. Consider $F^{i\leftrightarrow j}(K) := \left\{f_1^{i\leftrightarrow j}, \ldots, f_K^{i\leftrightarrow j}\right\}$ and the lottery $\frac{1}{K}\sum_{g\in F^{i\leftrightarrow j}(K)} TTC_{F^{i\leftrightarrow j}(K)}(\succ_{F^{i\leftrightarrow j}(K)}, SD_{F^{i\leftrightarrow j}(K)}(\succ); g)$. Since the positions of agent $i$ and $j$ are just reversed, the resulting lotteries are the same except that agent $i$ and $j$'s stochastic assignments are switched. That is, we have

$$\begin{aligned}&\frac{1}{K}\sum_{g\in F(K)} TTC_{F(K)}(\succ_{F(K)}, SD_{F(K)}(\succ); g)(i)\\&\quad=\frac{1}{K}\sum_{g\in F^{i\leftrightarrow j}(K)} TTC_{F^{i\leftrightarrow j}(K)}(\succ_{F^{i\leftrightarrow j}(K)}, SD_{F^{i\leftrightarrow j}(K)}(\succ); g)(j).\end{aligned}$$

Now, there exist nonempty and disjoint sets $\mathcal{H}$ and $\mathcal{H}'$ such that $\mathcal{H}\cup\mathcal{H}'=\mathcal{F}(K)$ and for each $F(K)\in\mathcal{H}$, $F^{i\leftrightarrow j}(K)\in\mathcal{H}'$. Then, using the above equation and letting $\varphi(F(K), g)=\frac{1}{K}TTC[\succ_{F(K)}, SD_{F(K)}(\succ); g]$,

$$\begin{aligned}TRSD^K(\succ)(i)&\equiv\frac{1}{\binom{n!}{K}}\sum_{F(K)\in\mathcal{F}(K)}\sum_{g\in F(K)}\varphi(F(K), g)(i)\\&=\frac{1}{\binom{n!}{K}}\sum_{F(K)\in\mathcal{F}(K)}\sum_{g\in F^{i\leftrightarrow j}(K)}\varphi(F^{i\leftrightarrow j}(K), g)(j)\\&=\frac{1}{\binom{n!}{K}}\sum_{F(K)\in\mathcal{H}}\sum_{g\in F^{i\leftrightarrow j}(K)}\varphi(F^{i\leftrightarrow j}(K), g)(j)\\&\quad+\frac{1}{\binom{n!}{K}}\sum_{F(K)\in\mathcal{H}'}\sum_{g\in F^{i\leftrightarrow j}(K)}\varphi(F^{i\leftrightarrow j}(K), g)(j)\\&=\frac{1}{\binom{n!}{K}}\sum_{F(K)\in\mathcal{H}'}\sum_{g\in F(K)}\varphi(F(K), g)(j)\\&\quad+\frac{1}{\binom{n!}{K}}\sum_{F(K)\in\mathcal{H}}\sum_{g\in F(K)}\varphi(F(K), g)(j)\\&=\frac{1}{\binom{n!}{K}}\sum_{F(K)\in\mathcal{F}(K)}\sum_{g\in F(K)}\varphi(F(K), g)(j)=TRSD^K(\succ)(k).\end{aligned}$$

The equality of the first term in the second and third line comes from the following: $[F(K)\in\mathcal{H}$ and $g\in F^{i\leftrightarrow j}(K)]\Leftrightarrow[F^{i\leftrightarrow j}(K)\in\mathcal{H}'$ and $g\in F^{i\leftrightarrow j}(K)]\Leftrightarrow[F'(K)\in\mathcal{H}'$ and $h\in\mathcal{H}']$. Similarly, the equality of the second term in the second and third line comes from the following: $[F(K)\in\mathcal{H}'$ and $g\in F^{i\leftrightarrow j}(K)]\Leftrightarrow[F^{i\leftrightarrow j}(K)\in\mathcal{H}$ and $g\in F^{i\leftrightarrow j}(K)]\Leftrightarrow[F'(K)\in\mathcal{H}$ and $h\in F'(K)]$. Hence, the TRSD$^K$ satisfies the equal treatment of equals. □

Part (1) Note that to show that $TRSD^K$ stochastically dominates $RSD$, we need to show that for some problem $\succ$, $TRSD^K\neq RSD$ due to the weakly stochastic dominance just proved above. If $|N|\leq 3$, then $RSD$ is sd-efficient (Bogomolnaia and Moulin 2001), and thus $TRSD^K=RSD$. Suppose $|N|\geq 4$. Example 3 shows that for $|N|=4$, $TRSD^K\neq RSD$. The extension to the case of $|N|\geq 5$ is straightforward.

Part (2) Suppose RSD is not sd-efficient for some $N$, $A$, and $q$. Then there is a problem $\succ$ such that $RSD(\succ)$ is not sd-efficient at $\succ$. Let $R := RSD(\succ)$. We first show

*Claim 2* there exist $\bar{K} \leq n$ and $F(\bar{K}) := \{f_1, \ldots, f_{\bar{K}}\}$ such that $SD_{F(\bar{K})}(\succ)$ is not Pareto efficient in the $|K|$-fold replica problem. First consider the case where $R$ is wasteful at $\succ$. Then there is $i \in N$, $a \in A$ with $r_{i,a} > 0$, and $b \in A$ with $b \succ_i a$ such that $\sum_{j \in N} r_{j,b} < q_b$. Then there is $f_1, f_2 \in F$ such that $SD_{f_1}(\succ)(i) = a$ and $\sum_{j \in N} |SD_{f_2}(\succ)(j)| < q_b$. Take $\bar{K} = 2$ and $F(\bar{K}) := \{f_1, f_2\}$. Then $SD_{F(\bar{K})}(\succ)$ is not Pareto efficient at $\succ$. Consider another case where $R$ is non-wasteful but not sd-efficient at $\succ$. Then there is a stochastic assignment $P$ such that $P$ stochastically dominates $R$ at $\succ$. Then, by Lemma 5, there is an improvement cycle $(a^1, i^1, a^2, i^2, \ldots, a^m, i^m, a^{m+1})$ from $R$ to $P$. Let $\bar{K} := m$. Then, since $a^1, \ldots, a^m$ are distinct, we have $\bar{K} \leq |A|$. Moreover, since $r_{i^\ell, a^\ell} > 0$ for each $\ell \in \{1, \ldots, m\}$, there is $F(\bar{K}) := \{f_1, \ldots, f_{\bar{K}}\}$, where $F(\bar{K})$ allows for duplicate elements, such that for each $\ell \in \{1, \ldots, m\}$, $SD_{f_k}(\succ)(i^\ell) = a^\ell$. Then an assignment $\nu$ where for each $i \in \{1, \ldots, m\}$, $\nu(i^\ell) = a^{\ell+1}$, Pareto dominates $SD_{F(\bar{K})}(\succ)$. Hence $SD_{F(\bar{K})}$ is not Pareto efficient at $\succ$. Thus the proof of Claim 2 is completed.

Let $K \geq \bar{K}$. Then there is $F(K) \subseteq F$ such that $F(\bar{K}) \subseteq F(K)$. Then, by Claim 2, $SD_{F(K)}(\succ)$ is not Pareto efficient. Thus, since $TRSD_{F(K)}(\succ_{F(K)}, SD_{F(K)}; g)$ for some $g \in F$ is Pareto efficient, we have $TRSD_{F(K)}(\succ_{F(K)}, SD_{F(K)}; g) \neq SD_{F(K)}(\succ)$. Therefore $TRSD^K \neq SD$.

## References


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

Abdulkadiroğlu A, Sönmez T (2003) Ordinal efficiency and dominated sets of assignments. J Econ Theory 112:157–172

Abdulkadiroğlu A, Sönmez T (2003) School choice: a mechanism design approach. Am Econ Rev 93:729–747

Athanassoglou S, Sethuraman J (2011) House allocation with fractional endowments. Int J Game Theory 40:481–513

Birkhoff G (1946) Three observations on linear algebra. Revi Univ Nac Tucuman Ser A 5:147–151

Bogomolnaia A, Moulin H (2001) A new solution to the random assignment problem. J Econ Theory 100:295–328

Budish E, Che Y-K, Kojima F, Milgrom P (2013) Designing random allocation mechanisms: theory and applications. Am Econ Rev 103:585–623

Che Y-K, Kojima F (2010) Asymptotic equivalence of random priority and probabilistic serial mechainsms. Econometrica 78:1625–1672

Erdil A (2014) Strategy-proof stochastic assignment. J Econ Theory 151:146–162

Gale D, Shapley LS (1962) College admissions and the stability of marriage. Am Math Monthly 69:9–15

Hashimoto T, Hirata D, Kesten O, Kurino M, Ünver MU (2014) Two axiomatic approaches to the probabilistic serial mechanism. Theor Econ 9:253–277

Hugh-Jones D, Kurino M, Vanberg C (2014) An experimental study on the incentives of the probabilistic serial mechanism. Games Econ Behav 87:367–380

Kesten O (2009) Why do popular mechanisms lack efficiency in random environments? J Econ Theory 144:2209–2226

Kesten O, Ünver MU (2015) A theory of school-choice lotteries. Theor Econ 10:543–595

Kojima F (2009) Random assignment of multiple indivisible objects. Math Soc Sci 57:134–142

Kojima F, Manea M (2010) Incentives in the probabilistic serial mechanism. J Econ Theory 144:106–123

Manea M (2008) Random serial dictatorship and ordinally efficient contracts. Int J Game Theory 36:489–496
Manea M (2009) Asymptotic ordinal inefficiency of random serial dictatorship. Theor Econ 4:165–197
Nesterov AS (2014) Fairness and efficiency in a random assignment. WZB Discussion Paper
Shapley L, Scarf H (1974) On cores and indivisibility. J Math Econ 1:23–37
von Neumann J (1953) A certain zero-sum two-person game equivalent to the optimal assignment problem. In: Kuhn HW, Tucker AW (eds) Contributions to the theory of games, vol 2. Princeton University Press, Princeton, New Jersey
Zhou L (1990) On a conjecture by Gale about one-sided matching problems. J Econ Theory 52:123–135