Kesten, Onur; Kurino, Morimitsu; Nesterov, Alexander

Working Paper

Efficient lottery design

WZB Discussion Paper, No. SP II 2015-203

Provided in Cooperation with:
WZB Berlin Social Science Center

Onur Kesten
Morimitsu Kurino
Alexander Nesterov

Efficient Lottery Design

Discussion Paper

SP II 2015–203

February 2015

(WZB) Berlin Social Science Center

Research Area
Markets and Choice

Research Unit
Market Behavior

Wissenschaftszentrum Berlin für Sozialforschung gGmbH
Reichpietschufer 50
10785 Berlin
Germany
www.wzb.eu

Onur Kesten, Morimitsu Kurino, Alexander Nesterov
Efficient Lottery Design

Affiliation of the authors:

Onur Kesten
Carnegie Mellon University

Morimitsu Kurino
University of Tsukuba

Alexander Nesterov
WZB

Wissenschaftszentrum Berlin für Sozialforschung gGmbH
Reichpietschufer 50
10785 Berlin
Germany
www.wzb.eu

Abstract

Efficient Lottery Design

by Onur Kesten, Morimitsu Kurino and Alexander Nesterov*

There has been a surge of interest in stochastic assignment mechanisms which proved to be theoretically compelling thanks to their prominent welfare properties. Contrary to stochastic mechanisms, however, lottery mechanisms are commonly used for indivisible good allocation in real-life. 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.

Keywords: Lottery; Ex post efficiency; sd–efficiency; random serial dictatorship

JEL classification: C71; C78; D71; D78

---

* E-mail: okesten@andrew.cmu.edu, kurino@sk.tsukuba.ac.jp, alexander.nesterov@wzb.eu.
We would like to thank 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.

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, which means that 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 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 a (sub)stochastic assignment matrix indicating agents' marginal assignment probabilities. In order 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. Two by now well-known examples of this approach are the *probabilistic serial* (PS) mechanism by Bogomolnaia and Moulin (2001) (BM hereafter) and *competitive equilibrium from equal incomes* (CEEI) mechanism by Hylland and Zeckhauser (1979),[2] both of which have become the cornerstones of a rapidly growing body of literature concerning stochastic mechanisms (cf. Kojima and Manea (2010), Yilmaz (2010), Hashimoto et al. (2014); Budish (2011); Kesten and Ünver (2011), He et al. (2012)).

BM have pointed out that the RSD outcome may suffer from unambiguous efficiency losses

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

[2]PS, which was originally proposed by Crès and Moulin (2001) for a simple model where all agents have identical preferences, 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 Section 5 for a more precise decription.

CEEI is a pseudo-market mechanism in which agents maximize utility subject to artificial budgets based on equal shares from the social endowments. It elicits cardinal preferences from agents and achieves ex ante efficiency, which is stronger than th sd-efficiency notion we study here.

regardless of the von Nemann-Morgenstern utilities compatible with agents' ordinal preferences. They 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-envy-freeness) property have triggered much interest to further extend and generalize PS to richer and more structured assignment problems (cf. Kojima 2009; Yilmaz 2010; Athanassoglou and Sethuraman 2011; Budish et al. 2013).

Since implementing a stochastic mechanism requires to decompose it into a lottery (Birkhoff, 1946; von Neumann, 1953; Budish et al., 2013), we contend that this additional procedure might hinder the transparency and practicality of such mechanisms. This is also the case when contrasting the two competing mechanisms, RSD and PS. Expressing their take on the feasible lottery mechanism RSD, Che and Kojima (2010) write:[3]

> *Perhaps more importantly for practical purposes, the random priority mechanism is straightforward and transparent, with the feasible lottery used for assignment specified explicitly. Transparency of a mechanism can be crucial for ensuring fairness in the eyes of participants, who may otherwise be concerned about "covert selection".*

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. On the other hand, the lottery approach has not been as successful as the stochastic approach as far as achieving stronger welfare properties than ex post efficiency.[4] Nevertheless, because a stochastic assignment needs to be decomposed into a feasible lottery before actual implementation, ex post considerations are comparably more difficult, if not impossible, to handle in the domain of stochastic assignments.[5] Therefore we believe that bridging the gap between the two approaches and developing tools which 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 directly done 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 (2003) 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. We provide a complementary result to this observation. In particular, we show that for any given stochastic assignment P of any

[3]The random serial dictatorship mechanism is also called the random priority.

[4]For example, as far as we are aware, a non-trivial 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).

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

given assignment problem $\succ$, there exists a corresponding deterministic assignment $\mu(P, \succ)$ which 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. A immediate corollary is the Abdulkadiroğlu and Sönmez's characterization of sd-efficiency to via notions of domination across sets of assignments.

An important contribution of our study, consistent with the commonly used methodology and trends in indivisible goods allocation literature,[6] is to develop a method for the construction of a lottery that improves upon a given inefficient lottery while maintaining feasibility of the final outcome (Theorem 2). 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 2). For the latter part of the objective we propose an algorithm that generates a feasible lottery from an infeasible lottery provided that it has a feasible equivalent. Section 4 provides an application of our tools and ideas, where we propose new lottery mechanisms that stochastically improve upon RSD. Our proposals combine the above mentioned methods with the celebrated object assignment method called *top trading cycles (TTC)* method of David Gale.[7] 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). We also provide practical approximations of TRSD that are computationally simpler (Theorem 4).

Finally, we offer a lottery mechanism that is equivalent to PS that may help facilitate its applicability in practice. The lottery representation of PS is based on the identification of a set

[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 good allocation. Examples abound. In housing markets (Shapley and Scarf, 1974), starting from an initial allocation of houses to agents, the objective, among other considerations, has been to achieve a Pareto efficient allocation that Pareto dominates the initial allocation. In this case Pareto domination of the initial allocation implies individual rationality. In the stochastic version of this model, a feasible lottery mechanism called *core from random endowments*, yields an ex-post efficient feasible lottery by individually Pareto improving upon each (distinct) feasible assignment in the support of a given ex-post inefficient initial feasible lottery. In on-campus housing, starting from a partial initial allocation of houses, a central objective has been again to obtain Pareto improvements while also using societal endowments of houses. The same objective has been pursued in kidney exchange problems with good samaritan donors (e.g., Sönmez and Ünver, 2006). In school choice with coarse priority structures, a common starting point in the recent literature (Erdil and Ergin, 2008; Abdulkadiroğlu et al., 2009; Kesten, 2010) has been the allocation induced by the lottery mechanism that uses the well-known *student-proposing deferred acceptance algorithm* upon randomly breaking the ties within equal priority classes. All these applications however have focused on achieving ex post properties.

[7]It was first proposed in the context of *housing markets* (Shapley and Scarf, 1974) where one seeks an optimal reallocation of objects (which are now endowments) among agents. 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* (Abdulkadiroğlu and Sönmez, 1999; Chen and Sönmez, 2002), *school choice* (Abdulkadiroğlu and Sönmez, 2003; Kesten, 2010), and *kidney exchange* (Roth et al., 2004, 2005). 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 and its relation to sd-efficiency for that matter. Kesten (2009) 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.

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 random assignment as that produced by the eating algorithm that yields 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, are constructed based on agents' preferences. This implies that in order to implement PS in this fashion 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, etc. We generalize this approach by proposing a lottery representation algorithm which, for any given random assignment, generates an equivalent equal-weight lottery (Theorem 5).

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 Section 5 the lottery representation of PS. Section 6 concludes.

2 The Model

A discrete resource allocation problem (Hylland and Zeckhauser, 1979; Shapley and Scarf, 1974) 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; $(q_a)_{a\in A}$ 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 $\succ_i$. Let $\mathbf{P}$ be the set of all preferences of any agent, and $\mathbf{P}^N$ the set of all preference profiles. The null object, if assumed to exist, is denoted by a_0 and 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 $\succ \in \mathbf{P}^N$.

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 $\mathcal{D}^f$ the set of all feasible assignments. A feasible assignment μ 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 $\sum_{b\in A} p_{i,b} = 0$. 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) The 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;

(L4) for each $s \in S$, $\mu_s \in \mathcal{D}$,

where w_s is called the **weight** of μ_s, and $\mu_S = (\mu_s)_{s\in S} \in \mathcal{D}^S$ is the **support** of L. Moreover, it has the **equal weights** if for each $s \in S$, $w_s = 1/|S|$. Also, it is **feasible** if instead of (L3), it satisfies (L3'): 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 μ. 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.

When we analyze a lottery in a later section, we look at some special "equivalent" lottery, where we mean 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.*

Proof. 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. $\square$

Example 1. Let $S = \{1, 2\}$, $\mu_1, \mu_2 \in \mathcal{D}^f$. Consider two lotteries $L = \frac{2}{3}\mu_1 + \frac{1}{3}\mu_2$ and $L' = \frac{1}{3}\mu_1 + \frac{1}{3}\mu_1 + \frac{1}{3}\mu_2$. Lottery L' is the equivalent equal-weight lottery of L. $\lozenge$

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

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

2.1 Axioms

A feasible lottery is **ex-post efficient** if it can be represented as a probability distribution over Pareto-efficient feasible assignments. A popular ex-post efficient feasible lottery mechanism is the random serial dictatorship (RSD). BM propose an appealing ex ante notion of sd-efficiency which also implies ex post efficiency, which we introduce next. Let $\succ \in \mathbf{P}^N$ be given. For each $i \in N$ and each $a \in A$, let $U(\succ_i, a) := \{b \in A \mid b \succeq_i a\}$ be the **upper contour set** of i at a. Given $i \in N$ and $P, R \in \mathcal{S}$, P_i **stochastically dominates** R_i at $\succ_i$ if for each $a \in A$, $\sum_{b \in U(\succ_i, a)} p_{i,b} \geq \sum_{b \in U(\succ_i, a)} r_{i,b}$. 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 $\sum_{j \in N} p_{j,b} = q_b$.

We define our fairness axiom. Let $\succ \in \mathbf{P}^N$. A stochastic assignment $P \in \mathcal{S}$ satisfies the **equal treatment of equals** at $\succ$ if for each $i, j \in N$, $\succ_i = \succ_j$ 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 ϕ is **sd-strategy-proof** if for each $\succ \in \mathbf{P}^N$, each $i \in N$, and each $\succ_i \in \mathbf{P}$, $\varphi_i(\succ)$ 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** φ **weakly stochastically dominates a stochastic mechanism** ψ if for each $\succ \in \mathbf{P}^N$, $\varphi(\succ)$ weakly stochastically dominates $\psi(\succ)$. Also, a **stochastic mechanism** ϕ **stochastically dominates a stochastic mechanism** ψ if ϕ weakly stochastically dominates ψ and for some $\succ \in \mathbf{P}^N$, $\phi(\succ)$ stochastically dominates $\psi(\succ)$ at $\succ$. Similarly, we can define the stochastic dominance of a lottery mechanism by looking at its induced stochastic mechanism.

3 Sd-efficiency and Pareto efficiency

3.1 Characterization of sd-efficiency

Abdulkadiroğlu and Sönmez (2003) investigate a possible link between sd-efficiency and Pareto efficiency. In particular, they ask whether 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: it turns out that even if every feasible assignment in the support of every feasible lottery that can be induced by 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 $s \in S$, $N_s = \{1_s, \cdots, i_s, \cdots, n_s\}$ be the set of all agents indexed by s, and $N_S := \cup_{s=1}^{S} N_s$ 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.

Let $\succ$ be a problem and S be an index set. An $|S|-$**fold replica assignment** is a function $\nu_S : N_S \to A$ such that for each $a \in A$, $|\nu^{-1}(a)| \leq q_a|S|$. 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 $i_s \in N_s$, $\nu_s(i_s) = \nu(i_s)$. Thus, we denote $\nu_S = (\nu_s)_{s \in S}$. An $|S|-$fold replica assignment $\nu_S = (\nu_s)_{s \in S}$ is **feasible** if for each $s \in S$, $s-$replica assignment 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 a 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** μ_S is the $|S|-$fold replica assignment where for all $s \in S$, each agent $i_s \in N_s$ is assigned object $\mu_s(i)$. Conversely, given an $|S|-$fold replica assignment ν_S, the **support (of a lottery) induced by the** $|S|-$**fold replica assignment** ν_S is the support in which at each event $s \in S$, each agent $i \in N$ is assigned object $\nu_s(i_s)$. Note that a lottery with the 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.*

Proof. Let μ_S be an $|S|-$fold replica assignment, and ν_S be the support of a lottery induced by μ_S. Also, let $P := (1/|S|) \sum_{s \in S} \pi(\nu_s)$. To show that P is a stochastic assignment, we need to check that (i) for each $i \in N$, $\sum_{a \in A} p_{i,a} = 1$, and (ii) for each $a \in A$, $\sum_{i \in N} p_{i,a} \leq q_a$. Condition (i) results from the fact that μ_S is a function from N_S to A. On the other hand, condition (ii) results from the facts that the lottery is of equal weights, and by definition of $|S|-$fold replica assignment, for each $a \in A$, $\sum_{s \in S} |\nu_s^{-1}(a)| \leq q_a|S|$.

□

Remark 1. By Lemma 2, from now on, unless confusion arises, we take the following: the support of an equal-weight lottery is an $|S|-$fold replica assignment, and vice versa.

An $|S|-$fold replica assignment μ_S **Pareto dominates** an $|S|-$fold replica assignment μ'_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 a problem $\succ_S$ 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 the induced support.

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

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

Theorem 1. *Let $\succ \in \mathbf{P}^N$ and L be 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. Whereas determining whether a random assignment is stochastically dominated or not may be difficult to check, Pareto efficiency of the support of a lottery is fairly straightforward by drawing on the top trading cycles (TTC) method which we later describe.[10]

3.2 Welfare improvement from an ex-post efficient lottery

In later sections, we aim to show that ex ante efficiency analysis besides ex post analysis can be directly done using lotteries. In particular, we shall propose a method to directly construct a new feasible lottery that stochastically improves upon a given sd-inefficient feasible lottery. 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 2. (Ex ante welfare improvement over an ex-post efficient lottery results in 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.

[10]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.

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

Consider the following ex-post efficient lottery.

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

Next consider the following feasible lottery:

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

Clearly, lottery L' stochastically dominates lottery L. However, the support of L' contains Pareto inefficient assignment μ_2. Thus L' is not ex-post efficient. We can show that there is no other feasible lottery that induces the stochastic assignment $\pi(L')$.[11] $\Diamond$

Given that sd-efficiency implies ex post efficiency, the observation in Example 2 is counter intuitive. It implies that ex post efficiency is not preserved under welfare improvements in stochastic assignments. Given an ex-post efficient but sd-inefficient feasible lottery, one can arrive at an sd-efficient (and thus ex-post efficient) feasible lottery by iteratively obtaining a stochastically improving lotteries (See Section 4).

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

[11]For the proof, we first show that there is no feasible and equivalent lottery with L' that is ex-post efficient and of the support size of 2. Let $L'' = w_1\mu_1'' + w_2\mu_2''$ where μ_1'' and μ_2'' are feasible. Consider agent 1. Since $\pi_1(L'') = (1/2, 0, 1/2, 0, 0)$, $w_1 = w_2 = 1/2$. Wlog, $\mu_1''(1) = a$ and $\mu_2''(1) = c$; Consider agent 2. Since $\pi_2(L'') = (0, 1/2, 1/2, 0, 0)$, either $[\mu_1''(2) = b$ and $\mu_2''(2) = c]$ or $[\mu_1''(2) = c$ and $\mu_2''(2) = b]$. Since $\mu_2''(1) = c$, by the feasibility of μ_2'', the former is impossible. Thus, $\mu_1''(2) = c$ and $\mu_2''(2) = b$; Consider agent 3. Since $\pi_3(L'') = (0, 1/2, 0, 1/2, 0)$, either $[\mu_1''(3) = b$ and $\mu_2''(3) = d]$ or $[\mu_1''(3) = d$ and $\mu_2''(3) = b]$. Since $\mu_2''(2) = b$, by the feasibility of μ_2'', the latter is impossible. Thus, $\mu_1''(3) = b$ and $\mu_2''(3) = d$; Consider agent 4. Clearly Thus, $\mu_1''(4) = d$ and $\mu_2''(4) = a$. Therefore, $\mu_1 = \mu_1''$ and $\mu_2 = \mu_2''$. Similarly, we can show that any equal-weight lottery that is equivalent to L' has the support consisting only of μ_1 and μ_2. Also, for any lottery, by Lemma 1 and its proof, there is its equivalent lottery L'' that only contains the same assignment as the original lottery. Thus, any feasible lottery that is equivalent to L' has the support consisting only of μ_1 and μ_2, and thus is not ex-post efficient.

post efficiency. To this end, we first take an equal-weight lottery with support μ_S equivalent to L (Lemma 1), and then by the correspondence of the support and $|S|-$fold replica assignment (Remark 1), we consider a Pareto improvement from μ_S in the $|S|-$fold replica problem. To this end, we introduce an $|S|-$**fold replica problem with endowments** μ_S where each agent $i_s \in N_S$ owns object $\mu_S(i_s) \equiv \mu_s(i_s)$. The problem is defined by $(\succ_S, \mu_S)$.

In order to have a Pareto improving assignment in the problem $(\succ_S, \mu_S)$, we use the notion of **(improvement) cycle**: it is a finite list of objects and agents $(a^1, i^1, a^2, i^2, \ldots, a^m, i^m)$, where $a^{m+1} := a^1$, such that each agent i^l owns object a^l, and prefers a^{l+1} to his owned object a^l. Finding an improvement cycle, and exchanging each agent's owned object for the next agent's owned object in the cycle, we can obtain a Pareto improving assignment.

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 Section 3.3, we propose a method that transforms a given infeasible lottery into an equivalent feasible one, and then in Section 4.3, we introduce a method of Pareto improvement in the replica problem with endowments.

3.3 Feasible assignment generating (FAG) algorithm

Given an equal-weight but infeasible lottery with the support $\mu_S = (\mu_s)_{s \in S}$, we introduce an algorithm that generates an equivalent and feasible lottery. To this end, we introduce some notion: an $|S|-$fold replica assignment μ_S is **frequency equivalent** to an $|S|-$fold replica assignment ν_S if for each $a \in A$, $|\mu_S^{-1}(a)| = |\nu^{-1}(a)|$. Note that as we defined in Section 3.1, an $|S|-$fold replica assignment ν_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 $S = \{1, 2, \ldots, |S|\}$. We focus on swapping objects in the set $\bar{A} := \left\{a \in A \mid |\mu_S^{-1}(a)| > 0\right\}$ — those which are assigned under μ_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 $\mu_S(S, i) = \left\{\mu(s, i) \in \bar{A} \mid s \in S\right\}$ and $\mu_S(1, I) = \left\{\mu(1, i) \in \bar{A} \mid i \in I\right\}$. Given $O \subseteq \bar{A}$, let

$$
\begin{aligned}
B(O) &= \cup_{i \in N:\, \mu(1,i) \in O} \left\{\mu_S(S, i)\right\}, \\
B^t(O) &= \begin{cases} O & \text{if } t = 1, \\ B(B^{t-1}(O)) & \text{if } t \geq 2. \end{cases}
\end{aligned}
$$

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 agents more than its quota at μ_1 (If no such object exists, μ_1 is feasible and we are done.). Let $X = \{c \in \bar{A} \mid |\mu_1^{-1}(c)| \leq q_c - 1\}$, i.e., the set of objects that are only partially assigned to agents at μ_1 under μ_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 whose proof can be found in the Appendix.

Claim 1. 1) $B^0(\{a\}) \subseteq B^1(\{a\}) \subseteq B^2(\{a\}) \subseteq \ldots$;

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, $\{a\} \subsetneq B^1(\{a\}) \subsetneq \ldots \subsetneq B^t(\{a\})$.

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 μ_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 $\mu_S(k_{i_t}, i_t) := b_{t-1} = x$.

Iteration. Given the support μ_S, repeating Phases 1 & 2 at most $n-1$ times yields a new support μ_S^1 whose first index assignment, μ_1^1, is feasible. Thus, we have finalized the first index assignment. Next we obtain a new support μ_S^2, whose first index assignment coincides with that of μ_S^1, by iteratively applying Phases 1 & 2 to the sub-support obtained from μ_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 μ_S, the FAG algorithm produces a feasible $|S|-$fold replica assignment that is frequency equivalent to μ_S.*

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

Corollary 1. *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 represent a rational stochastic assignment by an equal-weight infeasible lottery in a straightforward way. Thus, as a corollary of Proposition 1, we have

Corollary 2. *Any rational stochastic assignment can be expressed as a feasible equal-weight lottery that induces it.*

Remark 2. Note that Corollary 2 gives a version of Birkhoff (1946); von Neumann (1953) when a stochastic assignment is restricted to be rational.

Example 3. (Finding feasible assignment)

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

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

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 μ_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 re-apply Phases 1 & 2 to the remaining two rows. Then it is not much 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 μ_1, μ_2, and μ_3 respectively.

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

$\Diamond$

3.4 An alternative proof of an sd-efficiency characterization

Based on Theorem 1, we next provide an alternative proof of Abdulkadiroğlu and Sönmez (2003)'s characterization of sd-efficiency. Their characterization is based on the following "domination" notion.

Definition 3. *A set of feasible assignments M' dominates a set of feasible assignments M if*

1. there exists a set of assignments $\overline{M}$ that is frequency equivalent to M' and,

2. there exists a one-to-one function $f : \overline{M} \to M$ such that

(a) for all $\mu \in \overline{M}$, μ Pareto dominates or is equal to $f(\mu)$, and

(b) there exists $\mu \in \overline{M}$ such that μ Pareto dominates $f(\mu)$.

We provide a relatively shorter and transparent proof of the main characterization result of Abdulkadiroğlu and Sönmez (2003):

Corollary 3. (Abdulkadiroğlu and Sönmez, 2003) Given a problem $\succ$ let feasible lottery L be an arbitrary decomposition of a stochastic assignment P. P is sd-efficient at $\succ$ if and only if each subset M of the support of lottery L is undominated.

Proof. ($\Rightarrow$) Suppose there is a set $M \subseteq S(L)$ that is dominated. Let $\overline{M}$ be the set in Definition 3 that is frequency equivalent to M, and for each $\mu \in \overline{M}$ let $f(\mu) \in M$ be the assignment that (weakly) Pareto dominates μ. Let $\Sigma(L)$ be the allocation induced by the support of L, and β be an allocation that would be induced if each $\mu \in M$ were to be replaced by $f^{-1}(\mu) \in \overline{M}$. Clearly, β Pareto dominates $\Sigma(L)$. By Theorem 1 lottery L (and thus P) is sd-inefficient.

($\Leftarrow$) Suppose that lottery L with support $S(L) \equiv \{\mu_1, \mu_2, \ldots, \mu_K\}$ is sd-inefficient at $\succ$. By Theorem 1 there is an an allocation $\beta : N^K \to H$ which Pareto dominates $\Sigma(L)$. Let $\overline{M} = \{\mu'_1, \mu'_2, \ldots, \mu'_K\}$ be the set of assignments inversely-induced by allocation β. Since β Pareto dominates $S(L)$, for each $k \in \{1, \ldots, K\}$, μ'_k is either equal to or Pareto dominates μ_k. By Proposition 1, applying the FAG algorithm to the set $\overline{M}$ yields a set of feasible assignments M' that is frequency equivalent to $\overline{M}$. Thus $S(L)$ is dominated.

In the remainder of the paper, we provide illustrative applications of the tools and ideas we have developed so far for constructing lottery mechanisms. $\square$

4 Lottery Mechanism 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, the RSD is not sd-efficient but only ex-post efficient. In this section, we propose two methods of improving upon RSD.

4.1 The random serial dictatorship mechanism (RSD)

To define the random serial dictatorship mechanism (RSD), we introduce some notions: 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 $\succ_{f(1)}$.

⋮

Step k: The kth 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.

Remark 3. RSD is known to be sd-strategy-proof, ex-post efficient, and satisfy the equal treatment of equals. However, it is wasteful (Erdil, 2014).

4.2 Efficient lottery construction (ELC) procedure

We shall propose a method, called the **efficient lottery construction (ELC)** procedure, to directly construct a new ex-post efficient and feasible lottery that stochastically dominates a given sd-inefficient feasible lottery: Let a preference profile $\succ$ and a feasible lottery L be given. Our method is as follows.

Stage 1: (Initialization). We obtain an equivalent equal-weight lottery, L^e (Lemma 1), with the support μ_S. We decide which assignments in the the support to improve on, which is given by a partition $\mathcal{S}$ of S.

Stage 2 (Improvement). For each $M \in \mathcal{S}$, we consider the $|M|$-fold replica problem $\succ_M$ with endowments μ_M, and then apply some Pareto improvement algorithm (to be introduced in the next subsection) which selects a Pareto improving assignment ν_M.
Stage 3 (FAG algorithm). For each $M \in \mathcal{S}$ and each ν_M, we apply the FAG algorithm (Section 3.3) and obtain a feasible $|M|-$fold replica assignment, ν_M^f.
Stage 4: (Improvement) For each $m \in M$, we consider the original problem $\succ$ with endowments ν_m^f, and then apply some Pareto improvement algorithm (to be introduced in the next subsection) which delivers a Pareto efficient assignment $\hat{\nu}_m$.
Stage 5 (New lottery). Take the equal-weight lottery $\frac{1}{|S|}\sum_{s\in S}\hat{\nu}_s$.

Note that stage 2 implies that the resulting equal-weight lottery stochastically dominates the original lottery L (Lemma 3). Moreover, as we saw in Section 3.2 that the welfare improvement does not always lead to ex-post efficiency, Stage 4 makes sure that the new lottery is ex-post efficient. This is summarized in the next result whose straightforward proof is omitted.

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

4.3 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 is originally for a housing market where each object is owned by only one agent. In contrast, we deal with replica problems with endowments where an object is owned by multiple agents. Thus, we adapt TTC idea to our problem as follows.

For a given problem $(\succ_S, \mu_S)$, the **TTC algorithm** induces an $|S|-$fold replica assignment:

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

Step 1: Each agent points to her favorite object. Each object points to all of its owners. There is at least one (improvement) cycle although several cycles might intersect. Choose at least one cycle where if chosen cycles intersect, removing them should be feasible. Each agent in a chosen 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 of the other objects stay the same.

Step k: Each remaining agent points to her favorite object among the remaining ones. Each remaining object points to all of its remaining owners. There is at least one cycle. Choose at least one cycle where if chosen cycles intersect, removing them should be feasible. Each agent in a chosen 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 of the other objects stay the same.

The algorithm terminates when all agents are assigned objects. Notice that when several cycles intersect, we do not specify a cycle selection rule in the above algorithm. Thus, we will specify a cycle selection rule later in using the TTC algorithm.

An $|S|-$fold replica assignment ν_S is **individually rational** at $\succ_S$ if for each $i_s \in N_S$, $\nu_S(i_s) \succeq_i \mu_S(i_s)$. Now we are ready to state:

Proposition 2. *For each $|S|-$fold replica problem with endowments $(\succ, \mu_S)$, the TTC algorithm induces an individually rational and Pareto efficient assignment at $\succ_S$.*

The proof is in the Appendix.

4.4 Mechanisms that stochastically dominates the random serial dictatorship mechanism

4.4.1 TTC-based random serial dictatorship (TRSD) mechanism

Using the ELC procedure, we construct an *sd-efficient* lottery mechanism among the ones that stochastically dominate RSD, and satisfies the equal treatment of equals. As RSD induces an equal-weight lottery with the index set F and the support $SD_F(\succ) := (SD_f(\succ))_{f\in F}$, we take F as a unique element of the partition in the ELC procedure. In other words, given a problem $\succ\in \mathbf{P}^N$, we apply the TTC algorithm to the $|F|-$fold replica problem with endowments, $(\succ_F, SD_F(\succ))$, to obtain the induced $|F|-$fold replica assignment. By Theorem 2, the resulting lottery is ex-post efficient and weakly stochastically dominates RSD. Since the equal treatment of equals should be satisfied as part of our objective, we develop a cycle selection rule in the TTC algorithm for this purpose. We first introduce some notion.

Given $f \in F$ and distinct agents $i^1, \dots, i^m, j^1, \dots, j^m \in N$, define the priority $f^{(i^1\leftrightarrow j^1, i^2\leftrightarrow j^2, \dots, i^m\leftrightarrow j^m)} \in F$ such that for each $l \in \{1, \dots, m\}$, the positions of i^l and j^l in f is switched and all of the other agents keep the same priorities as in f.

Moreover, for each $i \in N$, let $N(i) := \{j \in N \mid \succ_j = \succ_i\}$ be the set of agents who have the same preferences as i does. Let $\mathcal{N} := \{N(i) \mid i \in I\}$. Note that for each $i \in N$, $i \in N(i)$ and $\mathcal{N}$ is a partition of N.[12]

Finally, given $f_1, \dots, f_m \in F$ and $i^1, \dots, i^m$ such that $\mathcal{N} = \{N(i^1), \dots, N(i^m)\}$, we define

$$F^{(f_1,\dots,f_m)} = \Big\{(h_1, \dots, h_m) \in F^m \mid \text{ for each } l \in \{1, \dots, m\} \text{ and some } j^l \in N(i^l), \text{ for each } k \in \{1, \dots, m\}, h_k = f_k^{(i^1\leftrightarrow j^1, \dots, i^m\leftrightarrow j^m)}\Big\}.$$

[12]A partition of a set N is a collection of disjoint nonempty subsets of N whose union is N.

Note that $|F^{(f_1,f_2,\dots,f_m)}| = |N(i_1)| \times |N(i_2)| \times \dots |N(i_m)|$.

Now we are ready to introduce a cycle selection rule in the TTC algorithm that is based on an exogenous priority $g \in F$. Consider each step in the TTC algorithm.

Round 1: Each remaining agent points to her top choice among the remaining objects. *Each remaining object points to an arbitrary agent whose type has the highest priority among these owners's types according to $g \in F$.* Since each agent points to one object and each object points to one agent, there is at least one cycle and no two distinct cycles intersect. We call them **initial cycles**.

Round 2: For each initial cycle, we make additional cycles (if they exist) as follows. Take an initial cycle denoted by $(a^1, i^1_{f_1}, a^2, i^2_{f_2}, \dots, a^m, i^m_{f_m})$ where for each $l \in \{1, \dots, m\}$, $i^l \in N$ and $i^l_{f_l} \in N_F$. Note that $N(i^1), N(i^2), \dots, N(i^m)$ are disjoint, and for each $l \in \{1, \dots, m\}$, $a^l = SD_{f_l}(\succ)(i^l)$.

Pick $(h_1, \dots, h_m) \in F^{(f_1,\dots,f_m)}$. Let $j^1 \in N(i^1), \dots, j^m \in N(i^m)$ such that for each $k \in \{1, \dots, m\}$, $h_k = f_k^{(i^1 \leftrightarrow j^1, \dots, i^m \leftrightarrow j^m)}$. Then, for each $l \in \{1, \dots, m\}$, $SD_{f_l}(\succ)(j^l) = SD_{h_l}(\succ)(i^l)$. Thus, we have a cycle $(a^1, j^1_{h_1}, a^2, j^2_{h_2}, \dots, a^m, j^m_{h_m})$, called an **additional cycle**. In this way, we can construct $|N(i^1)| \times \dots \times |N(i^m)|$ cycles.

Round 3: For each initial cycle, we remove the initial cycle and all of the additional cycles.

The **TTC-based RSD mechanism (TRSD)** induced by a priority $g \in F$ is defined to be an equal-weight lottery with the support induced by the TTC algorithm specified in the above.

Therefore, we have the following.

Theorem 3. *Suppose $n \geq 4$. For each $g \in F$, the TTC-based RSD mechanism induced by g is sd-efficient, stochastically dominates the RSD, and satisfies the equal treatment of equals.*

Proof. The sd-efficiency follows from Theorem 1 and Proposition 2. The stochastic dominance follows from Theorem 2. We show the equal treatment of equals. Let $\succ \in \mathbf{P}^N$ and $i^1, j^1 \in N$ such that $\succ_{i^1} = \succ_{j^1}$. Then, $j^1 \in N(i^1)$. It is sufficient to show that in each step of the TTC algorithm, the number of removed cycles involving agents with type i^1 is equal to the one involving agents with type j^1. In a step when an agent with type i is involved in a cycle $(a^1, i^1_{f_1}, a^2, i^2_{f_2}, \dots, a^m, i^m_{f_m})$, by construction in Round 2, the number of removed cycles involving agents with type i_1 is $|N(i^2)| \times |N(i^3)| \times \dots \times |N(i^m)|$, which is equal to the one involving agents with type j^1. $\square$

However, TRSD is not practical, because the size of the support is $|F| = n!$ which is computationally demanding as n gets large. In the next subsection, we develop a much more practical mechanism.

4.4.2 TTC-based random serial dictatorship^K mechanism (TRSD^K)

Using ELC procedure (Section 4.2), we introduce a practical lottery mechanism, called TTC-based random serial dictatorship^K mechanism (TRSD^K), which stochastically dominates RSD and satisfies the equal treatment of equals, where $K \in \{1, \ldots, n!\}$. To this end, we introduce a notation: Let

$$\mathcal{F}(K) = \{\{f_1, \ldots, f_K\} \mid f_1, \ldots, f_K \in F \text{ are distinct }\}.$$

Note that $|\mathcal{F}(K)| = \binom{n!}{K}$, where $\binom{n!}{K}$ is the number of $K-$combinations from $n!$ elements. We define the **TTC-based random serial dictatorship^K mechanism (TRSD$^{\boldsymbol{K}}$)**, where $K \in \{1, \ldots, n!\}$, by specifying the index set S and a partition $\mathcal{S}$ of S needed to use our ELC procedure as follows.

$$(\{f_1, \ldots, f_K\}, g, f) \in F(K) \times \{f_1, \ldots, f_K\} \times \{f_1, \ldots, f_K\} \in S,$$
$$\{(\{f_1, \ldots, f_K\}, g, f) \mid \{f_1, \ldots, f_K\} \in F(K),\, g, f \in \{f_1, \ldots, f_K\}\} \in \mathcal{S}.$$

We fix $\succ \in \mathbf{P}^N$ and $K \in \{1, \ldots, n!\}$. TRSD^K is the ELC procedure where the second stage we use TTC for the corresponding replica problem with endowments, and the selection rule in TTC is to use priority g. In particular,
Round 1: Choose $F(K) = \{f_1, \ldots, f_K\} \in \mathcal{F}(K)$ with equal probability $1/\binom{n!}{K}$.
Round 2: Choose $g \in F(K)$ with equal probability $1/K$.
Round 3: Consider the $|F(K)|-$fold replica problem with endowments $(\succ_{F(K)}, SD_{F(K)}(\succ))$, where $SD_{F(K)}(\succ) := (SD_f(\succ))_{f \in F(K)}$, and apply the following cycle selection rule in the TTC algorithm: In each step of the TTC algorithm, each remaining agent point to her favorite object among the remaining ones. Each remaining object points to an arbitrary agent whose type has the highest priority among these owners' types according to $g \in F(K)$. There is at least one cycle, and no two cycles intersect.

We apply the TTC algorithm with the above cycle selection given a priority $g \in F(K)$, and obtain the $|F(K)|-$fold replica assignment, denoted by $TTC_{F(K)}(\succ_{F(K)}, SD_{F(K)}(\succ); g)$. Its $f-$replica assignment is denoted by $TTC_f(\succ_{F(K)}, SD_{F(K)}(\succ); g)$. Then for each $f \in F(K)$, we apply the TTC algorithm for the original problem with endowments $TTC_f(\succ_{F(K)}, SD_{F(K)}(\succ); g)$. We denote this outcome by the same notation as $TTC_f(\succ_{F(K)}, SD_{F(K)}(\succ); g)$ for notational simplicity. Then, we run the induced equal-weight lottery $\frac{1}{K}\sum_{f \in F(K)} TTC_f(\succ_{F(K)}, SD_{F(K)}(\succ); g)$. In summary, the TRSD^K mechanism is formally expressed as

$$TRSD^K(\succ) = \frac{1}{\binom{n!}{K}} \sum_{F(K) \in \mathcal{F}(K)} \frac{1}{K} \sum_{g \in F(K)} \frac{1}{K} \sum_{f \in F(K)} TTC_f(\succ_{F(K)}, SD_{F(K)}(\succ); g). \tag{1}$$

Remark 4. Note that the TRSD^1 mechanism coincides with the RSD mechanism.

Theorem 4. *Let $K \in \{2, \ldots, n!\}$. TRSD^K is ex-post efficient, stochastically dominates RSD, and satisfies the equal treatment of equals. However, it is not sd-efficient.*

Proof. The ex-post efficiency follows from Theorem 2. To see the stochastic dominance, we express RSD as follows. Given a problem $\succ$,

$$
\begin{aligned}
RSD(\succ) &= \frac{1}{n!}\sum_{f\in F} SD_f(\succ) \\
&= \frac{1}{n!\binom{n!-1}{K-1}}\sum_{f\in F}\sum_{F(K)\in\mathcal{F}(K)\text{ s.t.}f\in F(K)} SD_f(\succ) \quad \left(\because |\{F(K)\in\mathcal{F}(K)\mid f\in F(K)\}| = \tbinom{n!-1}{K-1}\right) \\
&= \frac{1}{K\binom{n!}{K}}\sum_{f\in F}\sum_{F(K)\in\mathcal{F}(K)\text{ s.t.}f\in F(K)} SD_f(\succ) \quad \left(\because \tbinom{n!}{K} = \tbinom{n!-1}{K-1}\times\frac{n!}{K}\right) \\
&= \frac{1}{K\binom{n!}{K}}\sum_{F(K)\in\mathcal{F}(K)}\sum_{f\in F(K)} SD_f(\succ) = \frac{1}{K^2\binom{n!}{K}}\sum_{F(K)\in\mathcal{F}(K)}\sum_{g\in F(K)}\sum_{f\in F(K)} SD_f(\succ) \\
&= \frac{1}{\binom{n!}{K}}\sum_{F(K)\in\mathcal{F}(K)}\frac{1}{K}\sum_{g\in F(K)}\frac{1}{K}\sum_{f\in F(K)} SD_f(\succ). \qquad (2)
\end{aligned}
$$

Comparing the expression (1) of TRSD^K with the one (2) of RSD, by Theorem 2, given $F(K)\in\mathcal{F}(K)$ and $g\in F(K)$, the $|F(K)|-$fold replica assignment $TTC_{F(K)}(\succ_{F(K)}, SD_{F(K)}; g)$ Pareto dominates the one $SD_{F(K)}(\succ)$. Thus, by Lemma 3, TRSD^K stochastically dominates the RSD.

The proof for the equal treatment of equals can be found in the appendix. Moreover, we can see the sd-inefficiency by the computational simulation in Example 4. $\square$

Example 4. Let $N=\{1,2,3,4\}$, $A=\{a,b,a_0\}$, $q_a=q_b=1$, and $\succ\in\mathbf{P}^N$ such that

$\succ_1$	$\succ_2$	$\succ_3$	$\succ_4$
a	a	b	b
b	b	a	a
a_0	a_0	a_0	a_0

where a_0 is the null object. The following tables show the induced stochastic assignments through computer simulations.

RSD

	a	b	a_0
Agent 1	0.4167	0.0833	0.5000
Agent 2	0.4167	0.0833	0.5000
Agent 3	0.0833	0.4167	0.5000
Agent 4	0.0833	0.4167	0.5000

$TRSD^2$

a	b	a_0
0.4312	0.0688	0.5000
0.4312	0.0688	0.5000
0.0688	0.4312	0.5000
0.0688	0.4312	0.5000

$TRSD^3$

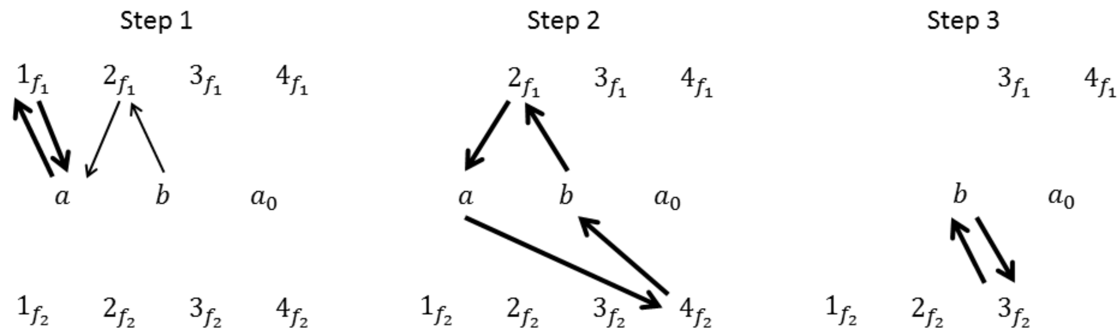
a	b	a_0
0.4417	0.0583	0.5000
0.4417	0.0583	0.5000
0.0583	0.4417	0.5000
0.0583	0.4417	0.5000

We show the working of TRSD^2.

Round 1: Suppose that $F(2)=\{f_1,f_2\}$ is drawn with probability $1/\binom{4!}{2}$ where $f_1=(1,2,3,4)$ and $f_2=(3,4,1,2)$. Then, $SD_{f_1}(\succ)=\begin{pmatrix}1 & 2 & 3 & 4\\ a & b & a_0 & a_0\end{pmatrix}$ and $SD_{f_2}(\succ)=\begin{pmatrix}1 & 2 & 3 & 4\\ a_0 & a_0 & b & a\end{pmatrix}$.

Round 2: Suppose that f_1 is drawn with probability $1/2$.

Round 3: Then, we apply the TTC algorithm induced by f_1 as follows.



Here, for simplicity, we draw only the pointing arrows from agents who are pointed out by objects. Then,

$$\begin{aligned} TTC_{F(2)}\left(\succ_{F(2)}, SD_{f_1}(\succ), SD_{f_2}(\succ); f_1\right) &= \frac{1}{2}\begin{pmatrix} 1 & 2 & 3 & 4 \\ a & a & a_0 & a_0 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 1 & 2 & 3 & 4 \\ a_0 & a_0 & b & b \end{pmatrix} \\ &= \frac{1}{2}\begin{pmatrix} 1 & 2 & 3 & 4 \\ a & a_0 & b & a_0 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 1 & 2 & 3 & 4 \\ a_0 & a & a_0 & b \end{pmatrix} \quad \text{(FAG algorithm)} \end{aligned}$$

Similarly, if in round 2, f_2 is drawn with probability $1/2$, then we have the same lottery as the previous one. That is, in this example, the selection of priorities from $F(2)$ does not matter for the TTC algorithm. The next example shows the case where it does matter. $\Diamond$

Example 5. (Cycle selection matters) Let $N = \{1, 2, \ldots, 6\}$,$A = \{a, b, c, d\}$, $q_a = q_b = q_c = q_d = 1$, and $\succ \in \mathbf{P}^N$ such that

$\succ_1$	$\succ_2$	$\succ_3$	$\succ_4$	$\succ_5$	$\succ_6$
b	c	a	d	b	a
a	b	c	c	d	d
a_0	a_0	a_0	a_0	a_0	a_0

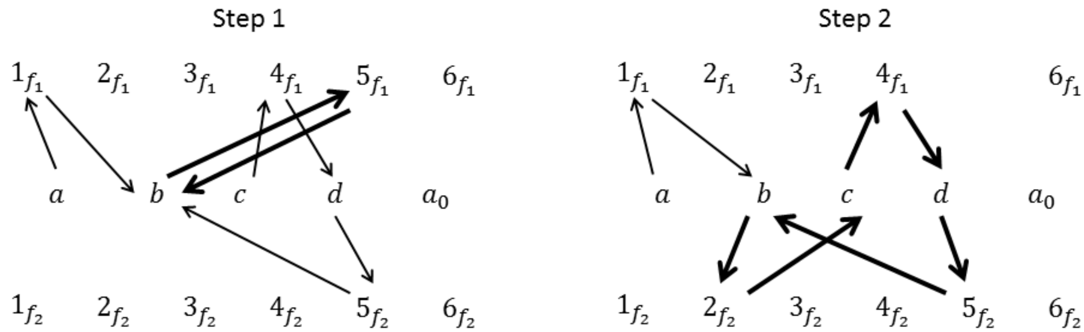
We consider $K = 2$.

Round 1: Suppose that $F(K) = \{f_1, f_2\}$ is drawn with probability $1/\binom{5!}{2}$ where $f_1 = (5, 1, 6, 4, \ldots)$ and $f_2 = (6, 3, 2, 5, \ldots)$. Then,

$$SD_{f_1}(\succ) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ a & a_0 & a_0 & c & b & d \end{pmatrix} \text{ and } SD_{f_2}(\succ) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ a_0 & b & c & a_0 & d & a \end{pmatrix}.$$

Round 2-case 1: Suppose that f_1 is drawn with probability $1/2$.

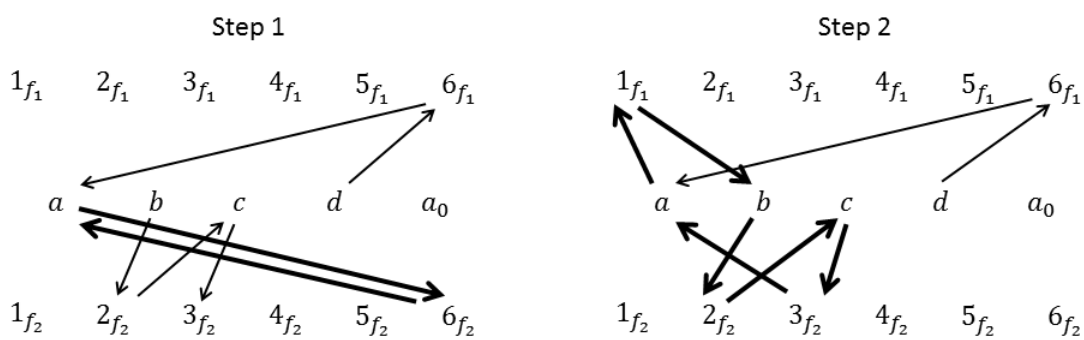
Round 3-case 1: Apply the TTC algorithm induced by f_1 as follows.



Here, to illustrate the difference of cycle selections, we illustrate the algorithm until a cycle selection matters. Then,

$$TTC_{F(2)}(\succ_{F(2)}, SD_{f_1}(\succ), SD_{f_2}(\succ); f_1) \;=\; \frac{1}{2}\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ a & a_0 & a_0 & d & b & d \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ a_0 & c & c & a_0 & b & a \end{pmatrix}.$$

<u>Round 2-case 2:</u> On the other hand, suppose that f_2 is drawn with probability 1/2.
<u>Round 3-case 2:</u> Apply the TTC algorithm induced by f_2 as follows.



Then,

$$TTC_{F(2)}(\succ, SD_{f_1}(\succ), SD_{f_2}(\succ); f_2) \;=\; \frac{1}{2}\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ b & a_0 & a_0 & c & b & d \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ a_0 & c & a & a_0 & d & a \end{pmatrix}.$$

$\Diamond$

5 A Lottery Representation of Probabilistic Serial

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 a method of representing the PS stochastic assignment by a lottery.

5.1 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:[13] 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 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 $\succ$.

Remark 5. PS is known to be weakly sd-strategy-proof, sd-efficient, and envy-free (Bogomolnaia and Moulin, 2001).

5.2 Lottery representation of the probabilistic serial mechanism

In this section we introduce an algorithm by which any PS stochastic assignment can be represented as an equal-weight lottery L. Specifically, for each preference profile $\succ$, we construct a set of priorities, F^*, such that

$$PS(\succ) = \frac{1}{|F^*|} \sum_{f \in F^*} SD_f(\succ).$$

Note that

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

where F is the set of all priorities. The difference is that F^* depends on preference profiles, while F^* might be smaller than F and will usually contain several copies of some of the priority orderings. We show how to construct F^*.[14]

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

[14] We can show a more general result in which any sd-efficient random assignment (and not only PS) can be represented as an equal-weight lottery using the same algorithm. The only requirement for such a stochastic assignment is that all its elements are rational numbers. It is easy to see that if some elements are irrational then the lottery representation is not feasible due to the fact that irrational weights are excluded by (L3) in the definition

Consider a preference profile $\succ \in \mathbf{P}^N$ and a random assignment $P = PS(\succ)$. Note that P is sd-efficient and contains only rational elements. Let us relabel the objects as $a_1, a_2, ..., a_k$, in the exhausting order in the eating algorithm of PS, that is, object a_1 is exhausted first, a_2 second, and so forth until a_k is exhausted in the end. When two or more objects are simultaneously exhausted we order them randomly. The objects which have not been eaten fully but only to some extent are put in the end of the ordering in some random order: $a_{k+1}, ..., a_{|A|}$. Note that the null-object a_0 is never exhausted and does not appear in this ordering. For each object a_j, let $E(a_j)$ be the set of agents that have eaten a_j, i.e., $E(a_j) = \{i \in N | p_{ia_j} > 0\}$.[15]

Since all elements in $P = \{p_{ij}\}$ are rational they can be represented as irreducible fractions of natural numbers $p_{ij} = \frac{r_{ij}}{t_{ij}}$, where $r_{ij}, t_{ij} \in \mathbb{N}$. Let t_j be the least common multiple among all t_{ij}, $i \in N$, corresponding to some object a_j:

$$t_j = \arg\min_{t'}(t' : \forall i \in N \;\; t' \vdots t_{ij}),$$

and in case some object $a_{j'}$ is not eaten at all, we assume its corresponding $t_{j'} = 1$.

We now construct a set of priorities $F^* = \{f_m\}_{m \leq |F^*|}$. For exposition purposes, we express the set F^* as a matrix (also denoted as F^*) whose elements are agents and whose columns determine priority orderings. Thus, the first row of the matrix F^*contains the agents that get top priorities, the second − agents with priorities of the second order and so forth till the last row which contains the agents that have bottom priorities.

Construction of the matrix F^* therefore requires determining the size and the elements of this matrix. In the algorithm however we first determine the elements, as if we knew the number of its columns $|F^*|$ already from the beginning. This assumption simplifies the algorithm and the notation and does not lead to any logical difficulties since we get the true size of the matrix as a result after we determine the elements in F^*.

In the algorithm, we proceed along the order of the objects. At each stage j of the algorithm we determine the j's row in the matrix F^* (denoted as F_j) and therefore find the j's position (starting from the top) in each priority $f \in F^*$. For example, in the first stage we find agents that have the top priority in all orderings, in the second − the second priority and so forth.

The matrix representation of F^* simplifies the procedure by which the set of priority orderings is constructed. This procedure is the following: at the first stage we divide the first row F_1 of the matrix F^* in several groups (the number of groups denoted as $|F_1|$) and all elements in each of these groups are assigned to some agent. (In terms of priority orderings, we partition the set F^*

of a lottery and that the sum of rational numbers is rational. Therefore, the algorithm can find an equal-weight lottery for any sd-efficient stochastic assignment if such a lottery exists. For the clarity of explanation we will use the case of PS, however the logic applies also for the general case.

[15] For the general case objects are also 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. Such profile of eating speed functions exists for each sd-efficient random assignment as shown in Bogomolnaia et al., 2001.

into several sets according to the top priorities in the priority orderings.)[16] In the second stage we divide *each* of the previous groups in several new smaller groups of equal size and each of these new groups is assigned one agent, and so forth. (Again, in terms of priority orderings, we partition each of the sets determined in the previous stage). Similarly, if at some stage j of the algorithm two columns f and f' of matrix F^* are said to belong to the same group F_{jm}, it means that the elements from the top down to slot j in these columns (and in the entire group F_{jm}) will be allocated identically, but the allocation below the row j might differ. (In terms of priority orderings, if orderings f and f' belong to the same partition F_{jm} of the set F^*, then positions in these orderings are assigned identically down to position j but they might differ after j.)

At each stage of the algorithm we exhaust one of the objects: a_1 is exhausted at the first stage, a_2 at the second and so forth, so that these objects are assigned to the agents with the corresponding priorities. The same rule applies to stages after stage k where objects are not exhausted anymore.

We can now define the algorithm formally.

Definition 4. The lottery representation algorithm constructs the set of priority orderings F^* by determining the elements in the corresponding matrix F^* in the following way:
Stage 1: Divide the first row F_1 of the matrix F^* into t_1 groups of equal size (such that each group has $\frac{|F^*|}{t_1}$ elements). In each of these groups assign elements of row F_1 to agents in $E(a_1)$ such that for each agent $i \in E(a_1)$ her share of elements in row F_1 is equal to her share of object a_1 in random assignment P: $p_{i1}|F^*| = |F_{1i}|$, where F_{1i} denotes the columns of matrix F^* that have agent i in their first row.[17]
Stage $j \leq k$: Consider first the groups defined at the previous stage $j-1$ in the row F_{j-1} of the matrix F^*. Divide *each* of these groups into q_j identical *subgroups* in the row F_j (we refer to them as to subgroups temporarily: at the next stage $(j+1)$ they become normal groups). If each agent $i \in E(a_j)$ appears in the procedure for the first time, then for each of the groups in F_{j-1} assign the elements in the row F_j of its subgroups to the agents in $E(a_j)$ according to their random assignment probabilities of a_j, as it was done at the stage 1. Otherwise, if some agent i' appeared in the earlier stages and received elements in some row $F_{j'}$ for some $j' < j$, then do the following:

(a). First assign this agent i' the total of $|F_j|p_{i'j}$ elements in the row F_j of those columns that did not include i' in the earlier stages, and do this assignment equally among such groups in F_{j-1}.[18]

[16] Notice again that during the procedure we determine the elements in matrix F^* without restricting its size $|F^*|$ from the beginning. That is why we may assume that the division operations for any of the rows at any stage are feasible (otherwise we could duplicate each column in matrix F^* enough times so that divisions become feasible).

[17] Notice that for the PS random assignment F_{1i} has exactly $\frac{|F^*|}{t_1}$ columns since all agents i in $E(a_1)$ gets the same share of a_1, which is no longer true in the general case.

[18] For example, let $p_{i'j} = \frac{1}{3}$, let there be four groups in F_{j-1}, each of them divided into six subgroups in F_j (thus twenty four subgroups in total), and in one of these four groups all columns contain agent i' at some earlier row. Then we first assign elements in the other three groups to agent i' (thus there are remaining eighteen subgroups to be potentially assigned to i'). We do it so that subgroups of i' are equally distributed among the three groups. Therefore, three subgroups in each of the three groups are assigned to agent i', which makes the share of elements in

Repeat this operation for all other such agents until no agent $E(a_j)$ remains that appeared at the earlier stages.

(c). Assign the elements in the remaining subgroups in F_j (corresponding to the remaining agents in $E(a_j)$) equally among the remaining groups in F_{j-1} such that every agent $i \in E(a_j)$ gets $|F^*|p_{ij}$ of the elements in F_j.

Stage $j > k$: Repeat the same procedure as for $j \leq k$ but assign some of the groups in F_j that correspond to the non-eaten part of a_j to the agents that prefer a_0 over any $a_{j'}$ where $j' \geq j$.

Final stage: Assign the remaining elements in matrix F^* to the remaining agents in such a way that no agent appears only once in each column. The weight of the equal-weight lottery is then $w = 1/|F^*|$.

The next example demonstrates the usage of the lottery representation algorithm.

Example 6. Let $|N| = 3, |A| = 4$ and the preference profile $\succ$ and the corresponding random assignment $PS(\succ)$ be as follows:

$$\begin{array}{ccc} \succ_1 & \succ_2 & \succ_3 \\ \hline a & a & b \\ b & c & a \\ c & b & a_0 \\ a_0 & a_0 & c \end{array}, P = \begin{pmatrix} 1/2 & 1/4 & 1/4 & 0 \\ 1/2 & 0 & 1/2 & 0 \\ 0 & 3/4 & 0 & 1/4 \end{pmatrix}.$$

The eating order is then a, b, c, where c is never fully eaten, therefore $k = 2$. The sets of eaters are: $E(a) = \{1, 2\}, E(b) = \{1, 3\}, E(c) = \{1, 2\}$. The least common multiples of the assignment probabilities are: $q_1 = 2, q_2 = 4$.

The matrix (and the set of priority orders) F^* is then found in $k + 1 = 3$ stages:

Stage 1: Divide the first row F_1 in F^* into $|F_1| = q_1 = 2$ equal groups F_{11} and F_{12}. Assign elements in F_{11} to agent $1 \in E(a)$ and elements of F_{12} to the other agent $2 \in E(a)$: for each $f \in F_{11}$, $f = (1, \ldots)$; and for each $f \in F_{12}$, $f = (2, \ldots)$.

Stage 2: In the second row of the priority matrix F^*, split the groups F_{11} and F_{12} into $q_2 = 4$ identical subgroups (eight subgroups in total, denote them as $f_m, m = 1, ..., 8$). Since agent 1 has appeared earlier at stage one while agent 3 did not, according to the rule (a) of the algorithm we first assign the subgroups to agent 1. At the previous stage we had only two groups F_{11} and F_{12} one of which was occupied by agent 1, therefore at the second stage agent 1 gets two subgroups (since $|F^*|p_{1b} = 2$) which belong to the group F_{12}. Let these subgroups be f_{25} and f_{26}. After this is done, assign the remaining slots to agent 3.

Stage 3: Complete the assignment in F^* by filling the third row F_3 with agents that have not appeared earlier in the priority orderings. The resulting priority set is then as follows:

F_j that correspond to i' and the share of object a_j in the random assignment P received by i' equal: $p_{i'j}|F^*| = |F_{ji'}|$.

$$F^* = \begin{pmatrix} F_1 \\ F_2 \\ F_3 \end{pmatrix} = \begin{pmatrix} F_{11}, & | & F_{12} \\ f_{21}, f_{22}, f_{23}, f_{24}, & | & f_{25}, f_{26}, f_{27}, f_{28} \\ f_{31}, f_{32}, f_{33}, f_{34}, & | & f_{35}, f_{36}, f_{37}, f_{38} \end{pmatrix} = \left\{ \begin{matrix} 1 & & 1 & & 1 & & 1 & & 2 & & 2 & & 2 & & 2 \\ 3 & , & 3 & , & 3 & , & 3 & , & 1 & , & 1 & , & 3 & , & 3 \\ 2 & & 2 & & 2 & & 2 & & 3 & & 3 & & 1 & & 1 \end{matrix} \right\}.$$

It is left to check that the equal-weight lottery over the support induced by these priorities is indeed identical to P. The following theorem proves this in general. $\Diamond$

Theorem 5. *For each preference profile $\succ \in \mathbf{P}^N$ and a random assignment P – containing only rational elements and 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 in at all stages of the algorithm and that it indeed induces an equal-weight lottery. These operations are: division of groups into subgroups and assignment of elements in the subgroups.

First, notice that at each stage j of the algorithm the equal division of the groups in F_{j-1} (into q_j subgroups in F_j) is feasible. This is due to the fact that all elements in P are rational numbers and that the dimension of F^* is made such that it is divisible by any natural number.

Second, after the equal division in F_j is done, each agent i receives $|F_j|p_{ij}$ of the groups in F_j. We first assign groups in F_j to those agents that appeared at earlier stages. Because of the nested structure of F^*, the only subgroups in F_j that cannot be assigned to i are those that belong to some group $F_{mj'}$ of an earlier stage j' that has been assigned to agent i already. (In terms of priority orderings, if for some subset of priority orderings agent i cannot be assigned the position j after the previous positions have been determined, it is only due to that i has been assigned some position in these priority orderings at earlier stages.) Since P is stochastic, there are at least $|F_j|p_{ij}$ subgroups in F_j that do not include elements assigned to i at earlier stages and thus these elements can be assigned to i at stage j. Therefore all stages are feasible and the algorithm induces an equal-weight lottery.

It is left to see that the resulting equal-weight lottery is equivalent to P. First we make sure that for each priority $f \in F^*$ each agent i that appears at some position $j \leq k$ (as well as in the row F_j and stage j of the algorithm) is indeed assigned object a_j under serial dictatorship given priority f. This is true since objects are initially ordered according to how they are exhausted during the simultaneous eating algorithm, and agents in this algorithm begin by eating their most preferred object and when it is exhausted continue to the next preferred among the remaining ones. Consequently, each agent i that appears at stage j prefers object a_j to any other object $a_{j'}$ unless $a_{j'}$ has appeared at an earlier stage and thus has been exhausted already. Therefore, under the serial dictatorship given priority f agent i picks a_j because it is her most preferred object among the remaining ones. For stages $j > k$ we can use the same logic for agents that receive normal objects and for agents that receive the null-object, since their more preferred objects have been already assigned and their best alternative among remaining ones is the null-object.

Finally, we make certain that the assignment probabilities in the lottery and in P match. For all objects the amount $|F^*|p_{ij}$ of elements in the row F_j is equivalent to the number of single priorities in F^* by construction of the groups. Therefore, since the weight in the equal-weight lottery is just $w = \frac{1}{|F^*|}$ agent i receives precisely $w|F^*|p_{ij} = p_{ij}$ of each object a_j. $\square$

The immediate corollary of the theorem is its application to PS, since its random assignment is always sd-efficient and contains only rational elements:

Corollary. *For each preference profile $\succ \in \mathbf{P}^N$, the lottery representation algorithm based on random assignment $P = PS(\succ)$ induces an equal-weight lottery that is equivalent to P.*

An important property of the lottery representation algorithm is that its induced lottery is flexible and can be further simplified or modified in order to achieve certain qualities.

First of all, this simplification can reduce the size of the support. For instance, for the preference profile used in the Example 2, for a PS stochastic assignment the algorithm gives the following set of priorities F^* (assuming objects are eaten as a, b, c, d) which can be reduced to the set F^{**} and even further to the set F^{***}:

$$F^* = \left\{ \begin{matrix} 1 \\ 3 \\ 2 \\ 4 \end{matrix}, \begin{matrix} 1 \\ 3 \\ 2 \\ 4 \end{matrix}, \begin{matrix} 1 \\ 4 \\ 2 \\ 3 \end{matrix}, \begin{matrix} 1 \\ 4 \\ 2 \\ 3 \end{matrix}, \begin{matrix} 2 \\ 3 \\ 1 \\ 4 \end{matrix}, \begin{matrix} 2 \\ 3 \\ 1 \\ 4 \end{matrix}, \begin{matrix} 2 \\ 4 \\ 1 \\ 3 \end{matrix}, \begin{matrix} 2 \\ 4 \\ 1 \\ 3 \end{matrix} \right\}, F^{**} = \left\{ \begin{matrix} 1 \\ 3 \\ 2 \\ 4 \end{matrix}, \begin{matrix} 1 \\ 4 \\ 2 \\ 3 \end{matrix}, \begin{matrix} 2 \\ 3 \\ 1 \\ 4 \end{matrix}, \begin{matrix} 2 \\ 4 \\ 1 \\ 3 \end{matrix} \right\}, F^{***} = \left\{ \begin{matrix} 1 \\ 3 \\ 2 \\ 4 \end{matrix}, \begin{matrix} 2 \\ 4 \\ 1 \\ 3 \end{matrix} \right\}.$$

Another possible modification deals with the distribution of slots in F^*. This can be done, for example, in order to make the lottery appear more equitable such that agents receive equal or comparable amounts of slots of different order: from top slots to bottom slots. Notice, that the priorities induced by the algorithm are flexible in that they allow for some replacements of the slots without changing the resulting assignment. For instance, in each of the priorities above one can replace the neighboring slots unless they are occupied by pairs of agents $\{1, 2\}$ or $\{3, 4\}$. Using these replacements in the second and third priority of F^{**} we can get the following equitable-looking set of priorities $\tilde{F^*}$:

$$\tilde{F^*} = \left\{ \begin{matrix} 1 \\ 3 \\ 2 \\ 4 \end{matrix}, \begin{matrix} 4 \\ 1 \\ 3 \\ 2 \end{matrix}, \begin{matrix} 3 \\ 2 \\ 4 \\ 1 \end{matrix}, \begin{matrix} 2 \\ 4 \\ 1 \\ 3 \end{matrix} \right\}.$$

Although the lotteries based on F^{**} and $\tilde{F^*}$ are essentially equivalent, the mechanism designer might prefer the latter for its fair appearance: under $\tilde{F}^*$ each of the four agents receives one of the four slots, while under F' agents 1 and 2 received the top slots and agents 3 and 4 received the bottom slots.

The question remains whether this result can be generalized: whether for any preference profile there exist an equal-weight lottery equivalent to the PS stochastic assignment such that each agent receives the same number of each slot in the priorities. Unfortunately, it is not the case as demonstrated by the following example.

Example 7. Let the preference profile $\succ$ be as follows:

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

The PS stochastic assignment and the set of priorities F^* are then as follows (here we assume that objects are eaten as a, b, c, d but another possible order b, a, c, d leads to the same result):

$$PS(\succ) = \begin{pmatrix} \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{3} & \frac{1}{6} \\ \frac{1}{2} & 0 & \frac{1}{3} & \frac{1}{6} \\ 0 & \frac{1}{2} & \frac{1}{3} & \frac{1}{6} \end{pmatrix}, F^* = \left\{ \begin{matrix} 1 \\ 2 \\ 4 \\ 3 \end{matrix}, \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix}, \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix}, \begin{matrix} 1 \\ 4 \\ 3 \\ 2 \end{matrix}, \begin{matrix} 1 \\ 4 \\ 3 \\ 2 \end{matrix}, \begin{matrix} 1 \\ 4 \\ 2 \\ 3 \end{matrix}, \begin{matrix} 3 \\ 2 \\ 4 \\ 1 \end{matrix}, \begin{matrix} 3 \\ 2 \\ 4 \\ 1 \end{matrix}, \begin{matrix} 3 \\ 2 \\ 4 \\ 1 \end{matrix}, \begin{matrix} 3 \\ 4 \\ 2 \\ 1 \end{matrix}, \begin{matrix} 3 \\ 4 \\ 2 \\ 1 \end{matrix}, \begin{matrix} 3 \\ 4 \\ 2 \\ 1 \end{matrix} \right\}.$$

Similarly to the previous example, we can adjust the first two lines in F^* such that the top two slots are equally distributed among all agents. However, the problem arises when we try to do the same with the next two lines. For example, agent 3 has four slots of level three and only two bottom slots which could be corrected by just one replacement, but this would distort the assignment. Indeed, in the priorities $\{f_2, ..., f_5\}$ that have to be corrected, agent 3 is followed either by agent 2 or agent 4. Since in all these priorities objects a, b have been picked by other agents already, agents in both pairs prefer object c to object d. Therefore replacing agent 3 by agent 2 or agent 4 distorts the resulting stochastic assignment. $\diamondsuit$

More generally, in terms of preferences the reason why the adjustment of slots in the example is not possible is the fact that agents 2, 3 and 4 have similar preferences for objects and receive identical assignment probabilities for objects c and d. Thus the resulting set of priorities F^* contains all three agents in rows three and four which makes the structure of F^* very tight and the necessary replacement becomes impossible. In order to generalize this observation, let us formally define the degree of heterogeneity of preferences and the degree of equitability of the equivalent lottery.

Definition 5. A preference profile $\succ$ is said to have a **degree of heterogeneity** k if in the object-agent correspondence $L^{\succ}$ induced by the top choice algorithm k is the minimal number of stages by which all agents appear at least once.

For PS, for instance, heterogeneity means that in the simultaneous eating algorithm the agents begin by eating different objects. The last exhausted object among all objects that agents began to eat as their top choices determines the degree of heterogeneity. In the examples above degree of heterogeneity of preferences equals two.

Definition 6. The equal-weight lottery is called **equitable of degree** $k \geq 0$ if its corresponding set of priorities F has identical number of slots for all agents from the top slot down to slot k.

Given these two definitions we formulate the following conjecture about the sufficient condition for the existence of an equitable lottery of degree k.

Conjecture 1. *For each preference profile $\succ$ and each stochastic assignment P (sd-efficient and containing only rational elements), there exist an equitable of degree k equal-weight lottery with stochastic assignment identical to P, where k is the level of heterogeneity of preferences in $\succ$.*

We are not able to prove the conjecture because of the complexity of the problem. The proof of the existence result in Theorem 5 is based on the nested structure of F^*, and this nested structure allows us to proceed from the top line of F^* to the bottom without violating any feasibility constraints or distorting the resulting random assignment. In case of the conjecture, on the contrary, in the process of modifying F via replacements of priority slots we need to proceed in both ways: from the top to the bottom and from the bottom to the top. Each replacement might restrict the set of remaining potential replacements, making the problem not very tractable.

If the conjecture is true, then a sufficient condition for a fully equitable-looking lottery (of degree n) would be the fact that some agent is assigned some object with certainty. In order to motivate the conjecture, we can demonstrate that the previous statement is true for Example 7 above if we add one more object e and one more agent 5 with opposing preferences such that the degree of heterogeneity becomes equal to five. The preference profile $\succ$ then writes as:

$\succ_1$	$\succ_2$	$\succ_3$	$\succ_4$	$\succ_5$
a	b	a	b	e
b	a	c	c	a
d	c	b	a	b
c	d	d	d	c
e	e	e	e	d

.

For this preference profile and the corresponding random assignment $P = PS(\succ)$ there exist an equal-weight lottery equitable of degree $n = 5$ with the following set of priority orderings $\tilde{F}^*$:

$$\left\{\begin{array}{l}
5,5,5,5,5,5,5,5,5,5,5,5,1,2,2,4,1,1,2,3,3,4,4,3,1,2,2,4,1,1,2,3,3,4,4,3,1,2,2,4,1,1,2,3,3,4,4,3,1,2,2,4,1,1,2,3,3,4,4,3\\
1,2,2,4,1,1,2,3,3,4,4,3,5,5,5,5,5,5,5,5,5,5,5,5,2,1,1,1,4,4,3,2,2,3,3,4,2,1,1,1,4,4,3,2,2,3,3,4,2,1,1,1,4,4,3,2,2,3,3,4\\
2,1,1,1,4,4,3,2,2,3,3,4,2,1,1,1,4,4,3,2,2,3,3,4,5,5,5,5,5,2,5,5,5,5,5,2,4,5,5,3,3,2,1,4,4,1,1,2,4,3,3,3,3,2,1,4,4,1,1,2\\
4,3,3,3,3,2,1,4,4,1,1,2,4,3,3,3,3,2,1,4,4,1,1,2,4,3,3,3,3,5,1,4,4,1,1,5,5,4,4,2,2,5,5,1,1,2,2,1,5,5,5,2,2,5,4,5,5,2,2,5\\
3,4,4,2,2,3,4,1,1,2,2,1,3,4,4,2,2,3,4,1,1,2,2,1,3,4,4,2,2,3,4,1,1,2,2,1,3,3,3,5,5,3,4,5,5,5,5,5,3,4,4,5,5,3,5,1,1,5,5,1
\end{array}\right\}.$$

As one this example demonstrates adding one agent with very different preferences provides enough flexibility in the set F^* such that it can be made equitable of degree n.

6 Concluding Remarks

In this paper we have introduced new tools that would allow the designer to directly work with lotteries and enhance the efficiency properties of an existing lottery mechanism. Whereas the stochastic approach has already proved extremely useful in achieving superior welfare features than their 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 on the construction of ex post and sd-efficient lotteries lends itself to new interpretations on the workings of the prominent mechanisms RSD and PS. Abdulkadiroğlu and Sönmez (1998) have shown 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 (as is the one induced by RSD). But because this procedure allows the trades to be carried out only within feasible assignments within the support of the initial lottery, the support of the new lottery may still admit an improvement cycle and thus the new lottery may be sd-inefficient (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 of this object.[19] Note that decomposing the initial stochastic assignment (that endows each agent with an equal probability of each object) leads to the lottery that that assigns an equal probability to each feasible assignment, i.e., the same initial lottery that the alternative lottery (to RSD) proposed by Abdulkadiroğlu and Sönmez (1998) relies on. The difference between the two mechanisms, however, comes 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

[19]See Kesten (2009) for a more precise description of this particular TTC procedure.

lottery individually, PS considers all the top trading cycles induced by all feasible assignments in the support of the initial lottery altogether.

In the U.S. many school districts use centralized clearinghouses to determine student assignments to public schools (see Abdulkadiroğlu and Sönmez 2003). 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. Priority orders may be determined based on different policy criteria such as walk zone, sibling status, special needs etc. 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 straightforwardly generalized and adapted to school choice problems, and in particular, could be helpful in improving the ex ante efficiency of school choice lotteries.

A Appendix

A.1 Proofs of Theorem 1

T prove Theorem 1, we need some notions and lemmas.

Definition 7. Let $\succ \in \mathbf{P}^N$ and $P, R \in \mathcal{S}$. An **improvement cycle from R to P** is a finite list $(a^1, i^1, a^2, i^2, \ldots, a^m, i^m)$, where $a^{m+1} \equiv a^1$ and $m \geq 2$, such that for each $l \in \{1, \ldots, m\}$, (i) $i^l \in N$ and $a^l \in A \setminus \{a_0\}$, (ii) $a^{l+1} \succ_{i^l} a^l$, (iii) $p_{i^l,a^l} < r_{i^l,a^l}$, and (iv) $p_{i^l,a^{l+1}} > r_{i^l,a^{l+1}}$.

Lemma 4. *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 $\succ$ and $P \neq R$. Then there is an improvement cycle from R to P.*

Proof. 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 $p_{i^1,a^1} < r_{i^1,a^1}$. Since $r_{i^1,a^1} > p_{i^1,a^1} \geq 0$, by *non-wastefulness* of R, $a^1 \succeq_{i^1} a^0$. Note that a^1 is not always in $A \setminus \{a_0\}$. Thus, since $a^2 \succ_{i^1} a^1$, $a^2 \in A \setminus \{a_0\}$. Also, by *feasibility* of P and *non-wastefulness* of R, since $r_{i^1,a^1} > p_{i^1,a^1} \geq 0$ and $a^2 \succ_{i^1} a^1$, we have $\sum_{j \in N} p_{j,a^2} \leq q_{a^2} = \sum_{j \in N} r_{j,a^2}$. Thus, since $p_{i^1,a^2} > r_{i^1,a^2}$, there is $i^2 \in N$ such that $p_{i^2,a^2} < r_{i^2,a^2}$. Thus, since P_{i^2} stochastically dominates R_{i^2} at $\succ_{i^2}$, there is $a^3 \in A$ such that $a^3 \succ_{i^2} a^2$ and $p_{i^2,a^3} > r_{i^2,a^3}$. Since $r_{i^2,a^2} > p_{i^2,a^2} \geq 0$, by *non-wastefulness* of R, $a^2 \succeq_{i^2} a_0$. Thus, since $a^3 \succ_{i^2} a^2$, $a^3 \in A \setminus \{a_0\}$. Also, by *feasibility* of P and *non-wastefulness* of R, since $r_{i^2,a^2} > p_{i^2,a^2} \geq 0$ and $a^3 \succ_{i^2} a^2$), we have $\sum_{j \in N} p_{j,a^3} \leq q_{a^3} = \sum_{j \in N} r_{j,a^3}$. Repeating this process, since N and A are finite, there is a cycle $(a^l, i^l, \ldots, a^m, i^m)$ where $1 \leq l \leq m-1$. Note that if $l = 1$, $a^1 = a^{m+1} \in A \setminus \{a_0\}$. $\square$

Proof of Theorem 1. Let L be a lottery with the support μ_S: ($\Rightarrow$) We show the contrapositive. Suppose that the support μ_S of L is not Pareto efficient at $\succ_S$. Then, there is an $|S|-$fold replica assignment ν_S that Pareto dominates μ_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 ν'_M: for $m \in M$, $\nu'_m = \nu_{s(m)}$. Then, ν'_M Pareto dominates μ'_M at $\succ_M$. By Lemma 3, the equal-weight lottery with the support ν'_M stochastically dominates the equal-weight lottery μ'_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 ν_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 μ_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 4, 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 ν_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 μ. Then, ν_S Pareto dominates μ_S at $\succ_S$. $\square$

A.2 Other proofs

Proof of Proposition 2. Let ν_S be an $|S|-$fold replica assignment induced by the TTC algorithm. Consider the TTC algorithm. We first show individual rationality. Each agent $i_s \in N_S$ is an owner of some object a. Object a points to her until she leaves. Therefore, the assignment of i_s cannot be worse than her owned object a.

We show Pareto efficiency. Suppose for a contradiction that there is an $|S|-$fold replica assignment ν'_S that Pareto dominates ν_S at $\succ_S$. Let N_S^k be the set of agents who leave at step k and K be the last step of the algorithm. We show by induction on k that for each $k = 1, 2, \ldots, K$ and each $i \in N_S^k$, $\nu'_S(i) = \nu_S(i)$. Any agent who leaves at Step 1 is assigned her top choice and thus cannot be made better off. Thus, for each agent $i \in N_S^1$, $\nu'_S(i) = \nu_S(i)$.

Suppose that the claim is true up to step $k-1$ where $k \geq 2$. Suppose for some agent $i \in N_S^k$, $\nu'_S(i) \neq \nu_S(i)$. Since ν'_S Pareto dominates ν_S at $\succ_S$, $\nu'_S(i) \succ_i \nu_S(i)$. Since $i \in N_S^k$ and $\nu'_S(i) \succ_i \nu_S(i)$, the object $\nu'_S(i)$ is already removed at an earlier step. That is, letting $a := \nu'_S(i)$, we have $|\{j \in C_1 \cup \ldots \cup C_{k-1} \mid \nu_S(j) = a\}| = q_a|S|$. By the induction hypothesis, $\{j \in C_1 \cup \ldots \cup C_{k-1} \mid \nu'_S(j) = a\}| = q_a|S|$. Thus, since $a = \nu'_S(i)$, we have $|(\nu'_S)^{-1}(a)| > q_a|S|$, contradicting the feasibility of ν'_S. $\square$

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 $\{i_1, \ldots, i_M\} := \{i \in I \mid \mu_S(1,i) \in B^t(\{a\})\}$, 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 μ_S.
Part (3): If the claim is not true, we have $\{a\} \subsetneq B^1(\{a\}) \subsetneq \ldots \subsetneq B^t(\{a\}) \subsetneq \ldots$, which contradicts the finiteness of A. $\square$

Proof of Theorem 4. We show that TRSD^K satisfies the equal treatment of equals. Let $i, j \in N$ with $i \neq j$ and $\succ \in \mathbf{P}^N$ such that $\succ_i = \succ_j$. 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. Agents face the equal-weight lottery $\frac{1}{K} \sum_{g \in F(K)} TTC_{F(K)}(\succ_{F(K)}, SD_{F(K)}(\succ); g)$. Then, consider $F^{i \leftrightarrow j}(K) = \left\{f_1^{i \leftrightarrow j}, \ldots, f_K^{i \leftrightarrow j}\right\}$ and the equal-weight 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 role of agent i and j is just reversed, the resulting lotteries are the same except that agent i and j's stochastic assignments are switched. That is, we have

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

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 $\phi(F(K), g) = \frac{1}{K} TTC[\succ_{F(K)}, SD_{F(K)}(\succ); g]$,

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

The equality of the first term in the second and the third line comes from the following: $[F(K) \in \mathcal{H} \text{ and } g \in F^{i \leftrightarrow j}(K)] \Leftrightarrow [F^{i \leftrightarrow j}(K) \in \mathcal{H}' \text{ and } g \in F^{i \leftrightarrow j}(K)] \Leftrightarrow [F'(K) \in \mathcal{H}' \text{ and } h \in \mathcal{H}']$.

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

References

Abdulkadiroğlu, A., P. A. Pathak, and A. E. Roth (2009): "Strategy-proofness versus Efficiency in Matching with Indifferences: Redesigning the NYC High School Match," *American Economic Review*, 99, 1954–1978.

Abdulkadiroğlu, A. and T. Sönmez (1998): "Random Serial Dictatorship and the Core from Random Endowments in House Allocation Problems," *Econometrica*, 66, 689–701.

——— (1999): "House Allocation with Existing Tenants," *Journal of Economic Theory*, 88, 233–260.

Abdulkadiroğlu, A. and T. Sönmez (2003): "Ordinal Efficiency and Dominated Sets of Assignments," *Journal of Economic Theory*, 112, 157–172.

Abdulkadiroğlu, A. and T. Sönmez (2003): "School Choice: A Mechanism Design Approach," *American Economic Review*, 93, 729–747.

Athanassoglou, S. and J. Sethuraman (2011): "House Allocation with Fractional Endowments," *International Journal of Game Theory*, 40, 481–513.

Birkhoff, G. (1946): "Three Observations on Linear Algebra," *Revi. Univ. Nac. Tucuman, ser A*, 5, 147–151.

Bogomolnaia, A. and H. Moulin (2001): "A New Solution to the Random Assignment Problem," *Journal of Economic Theory*, 100, 295–328.

Budish, E. (2011): "The Combinatorial Assignment Problem: Approximate Competitive Equilibrium from Equal Incomes," *Journal of Political Economy*, 119, 1061–1103.

Budish, E., Y.-K. Che, F. Kojima, and P. Milgrom (2013): "Designing Random Allocation Mechanisms: Theory and Applications," *American Economic Review*, 103, 585–623.

Che, Y.-K. and F. Kojima (2010): "Asymptotic Equivalence of Random Priority and Probabilistic Serial Mechainsms," *Econometrica*, 78, 1625–1672.

Chen, Y. and T. Sönmez (2002): "Improving Efficiency of On-campus Housing: An Experimental Study," *American Economic Review*, 92, 1669–1686.

Crès, H. and H. Moulin (2001): "Scheduling with Opting Out: Improving upon Random Priority," *Operations Research*, 49, 565–577.

Erdil, A. (2014): "Strategy-proof Stochastic Assignment," *Journal of Economic Theory*, 151, 146–162.

Erdil, A. and H. Ergin (2008): "What's the Matter with Tie-breaking? Improving Efficiency in School Choice," *American Economic Review*, 98, 669–689.

Gale, D. and L. S. Shapley (1962): "College Admissions and the Stability of Marriage," *American Mathematical Monthly*, 69, 9–15.

Hashimoto, T., D. Hirata, O. Kesten, M. Kurino, and M. U. Ünver (2014): "Two Axiomatic Approaches to the Probabilistic Serial Mechanism," *Theoretical Economics*, 9, 253–277.

He, Y., A. Miralles, and J. Yan (2012): "Competitive Equilibrium from Equal Incomes for Two-Sided Matching," Working paper.

Hugh-Jones, D., M. Kurino, and C. Vanberg (2014): "An Experimental Study on the Incentives of the Probabilistic Serial Mechanism," *Games and Economic Behavior*, 87, 367–380.

Hylland, A. and R. Zeckhauser (1979): "The Efficient Allocation of Individuals to Positions," *Journal of Political Economy*, 87, 293–314.

Kesten, O. (2009): "Why Do Popular Mechanisms Lack Efficiency in Random Environments?" *Journal of Economic Theory*, 144, 2209–2226.

——— (2010): "School Choice with Consent," *Quaterly Journal of Economics*, 125(3), 1297–1348.

Kesten, O. and M. U. Ünver (2011): "A Theory of School-Choice Lotteries," Forthcoming in *Theoretical Economics*.

Kojima, F. (2009): "Random Assignment of Multiple Indivisible Objects," *Mathematical Social Sciences*, 57, 134–142.

Kojima, F. and M. Manea (2010): "Incentives in the Probabilistic Serial Mechanism," *Journal of Economic Theory*, 144, 106–123.

Roth, A. E., T. Sönmez, and M. U. Ünver (2004): "Kidney Exchange," *Quarterly Journal of Economics*, 119, 457–488.

——— (2005): "Pairwise Kidney Exchange," *Journal of Economic Theory*, 125, 151–188.

Shapley, L. and H. Scarf (1974): "On Cores and Indivisibility," *Journal of Mathematical Economics*, 1, 23–37.

Sönmez, T. and M. U. Ünver (2006): "Kidney Exchange with Good Samaritan Donors: A Characterization," Working paper.

von Neumann, J. (1953): "A Certain Zero-sum Two-person Game Equivalent to the Optimal Assignment Problem," in *Contributions to the theory of games, Vol. 2*, ed. by H. W. Kuhn and A. W. Tucker, Princeton, New Jersey: Princeton University Press.

Yilmaz, Özgür. (2010): "The Probabilistic Serial Mechanism with Private Endowments," *Games and Economic Behavior*, 69, 475–491.

Zhou, L. (1990): "On a Conjecture by Gale about One-Sided Matching Problems," *Journal of Economic Theory*, 52, 123–135.

Discussion Papers of the Research Area Markets and Choice 2015

Research Unit: **Market Behavior**

Sebastian Kodritsch A note on the welfare of a sophisticated time-inconsistent decision-maker	SP II 2015-201
John Duffy, Dietmar Fehr Equilibrium selection in similar repeated games: Experimental evidence on the role of precedents	SPII 2015-202
Onur Kesten, Morimitsu Kurino, Alexander Nesterov Efficient lottery design	SPII 2015-203

WZB Junior Research Group: **Risk and Development**

Ferdinand M. Vieider, Clara Villegas-Palacio, Peter Martinsson, Milagros Mejía Risk taking for oneself and others: A structural model approach	SP II 2015-401

All discussion papers are downloadable:
http://www.wzb.eu/en/publications/discussion-papers/markets-and-choice