5-2009

Why Do Popular Mechanisms Lack Efficiency in Random Environments?

Onur Kesten
Carnegie Mellon University, okesten@andrew.cmu.edu

Published In
Journal of Economic Theory, 144, 5, 2209-2226.

Why do popular mechanisms lack efficiency in random environments?

Onur Kesten*

May 2009

Abstract

We consider the problem of randomly assigning n indivisible objects to n agents. Recent research introduced a promising mechanism, the *probabilistic serial* that has superior efficiency properties than the most common real-life mechanism *random priority*. On the other hand, mechanisms based on Gale's celebrated *top trading cycles* method have long dominated the indivisible goods literature (with the exception of the present context) thanks to their outstanding efficiency features. We present an equivalence result between the three kinds of mechanisms, that may help better understand why efficiency differences among popular mechanisms might arise in random environments. This result also suggests that the probabilistic serial and the random priority mechanisms can be viewed as two top trading cycles based mechanisms that essentially differ in the initial conditions of the market before trading starts.

JEL Classification: *C71; C78; D71; D78*

Key words: Indivisible goods; Random priority; Probabilistic serial; Top trading cycles; Ordinal efficiency

1 Introduction

The problem of assigning n indivisible objects to n agents where each agent receives exactly one object is usually known as the *assignment problem* (also referred as the *house allocation problem*). Typical examples are the assignment of houses to applicants, offices to graduate students, tasks to workers, and so on.

The deterministic approaches to this problem suffer from the conflict between fairness and efficiency. Therefore a common tool is to introduce randomization to recover compatibility between the two requirements. In this richer setting, the problem takes the name the *random assignment problem*. A *random*

*Tepper School of Business, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213, USA; Tel:+1-412-268-9823; Fax:+1-412-268-7064; e-mail: okesten@andrew.cmu.edu. I thank an associate editor of the journal as well as two anonymous referees for useful suggestions. I have also benefited from helpful discussions with Tayfun Sönmez, Al Roth, Steve Spear, William Thomson, Herve Moulin, Utku Unver, Mihai Manea, Ozgun Ekici and participants of the SAET 2007 conference (Greece) and seminar participants at the University of Montreal. Any remaining errors are my own.

assignment specifies a random allotment over objects for each agent. A *mechanism* is a systematic way of selecting a random assignment for any given random assignment problem.

Probably the most common real-life mechanism of this problem is the *random priority (RP)*:[1] A random ordering of agents is drawn from the uniform distribution, and agents, starting with the first agent, are asked to successively choose their favorite objects from the available ones. Differently put, RP is the induced average of $n!$ serial dictatorship outcomes. An important advantage of RP is that it only requires *ordinal* preferences of agents over deterministic outcomes as opposed to *cardinal* preferences over lotteries.[2] RP induces *ex post efficient* random assignments (i.e., it can be represented as a probability distribution over Pareto efficient deterministic assignments), and moreover selects the central point within this set.

Bogomolnaia and Moulin (2001) introduce and study a more appealing and an even stronger notion of efficiency than ex post efficiency which they call "ordinal efficiency:" A random assignment is *ordinally efficient* if it is not stochastically dominated by another random assignment. Surprisingly, widely-used RP may not always induce ordinally efficient outcomes. Bogomolnaia and Moulin (2001) characterize the entire set of ordinally efficient random assignments for any given random assignment problem. They propose an important contender to RP which selects the unique central point in the ordinally efficient set. They call this mechanism the *probabilistic serial (PS).*

An increasingly popular resource allocation method in the indivisible goods literature is the so called *top trading cycles (TTC)* method. It is attributed to David Gale, and 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 contexts which include the single and multiple *assignment problems*,[3] the *school choice problem,*[4] and the *kidney exchange problem*[5].

Abdulkadiroğlu and Sönmez (1998) propose a first adaptation of the TTC method to the present context, which they call, *core from random endowments (CfRE)*: A random initial assignment of objects is drawn from the uniform distribution, and the TTC method is applied to the resulting housing market. It turns out that RP and CfRE are equivalent (Abdulkadiroğlu and Sönmez, 1998). Although for deterministic settings, all TTC based mechanisms are efficient, the only probabilistic TTC based mechanism CfRE, by

[1] It is also referred as *random serial dictatorship.* Zhou (1990) is the first to study RP in the literature. Much earlier, Hylland and Zeckhauser (1979) proposed a mechanism that is based on a competitive equilibrium approach.

[2] See for example Kagel and Roth (1995).

[3] See for example, Svensson (1999), Abdulkadiroğlu and Sönmez (1999), Pápai (2000), and Ehlers (2002).

[4] The problem of assigning a set of students to a set of schools, where each school has a certain capacity, and a priority ordering of students. See for example, Abdulkadiroğlu and Sönmez (2003b).

[5] The problem of reallocating kidneys from a group of donors to a group of recipients. See for example, Roth, Sönmez, and Unver (2004).

this equivalence, may not induce ordinally efficient outcomes.

In this paper we aim to understand why popular real-life mechanism RP and mechanisms based on adaptations of the popular TTC method (such as CfRE) may be short of efficiency in a random assignment environment. This, we believe, is a key question the answer to which may prove quite useful for indivisible good allocation problems in general.

We propose an alternative mechanism based on approaching the random assignment problem from an individual ownership perspective: Initially all objects are collectively owned, and each agent a priori can be seen to have equal right over each object. This means (assuming n agents and n objects) that each agent should have the right to receive any object with probability $1/n$ and, suggests that each agent should initially be entitled to a $1/n$ share of each object. Such an interpretation enables one to view the random assignment problem similarly to a housing market in which one can again apply the TTC method in a simple and intuitive way allowing agents to uniformly carry out the trades in their best interest. We call this class of mechanisms *top trading cycles from equal division (TTCfED).*

Next we turn our attention to RP, and argue that its efficiency loss could be related with the degree of variation across individual serial dictatorship outcomes. To make this point, we introduce the following extension of the RP procedure: Consider an extended problem with a $k-$replica of resources, and compute the random assignment which is the induced average of $kn!$ serial dictatorship outcomes where for each ordering of agents the associated serial dictatorship is applied consecutively k times until all objects are exhausted. We call this class of mechanisms *random priority (RP)* $[*k]$.

Our main result is that the two classes of mechanisms TTCfED and RP $[*k]$ coincide with PS when one restricts attention to single and pairwise cycles for the former class, and when resources are replicated infinitely many times for the latter one (Theorem 1). This result allows us to better understand the differences and commonalities of the two competing mechanisms RP and PS. In particular, the first part of this result (i.e., the equivalence of PS and TTCfED) together with the result of Abdulkadiroğlu and Sönmez (1998) suggests that both PS and RP can be interpreted as top trading cycles mechanisms. Based on such an interpretation the basic difference between the two mechanisms essentially comes from how each mechanism chooses the initial conditions of the market before allowing agents to trade. On the other hand, the second part of Theorem 1 (i.e., the equivalence of PS and RP $[*k]$ as $k \to \infty$) confirms our earlier intuition about the efficiency loss of RP. Indeed, as the number of replicas of each object goes to infinite, the variation across individual serial dictatorship outcomes disappears thereby enabling ordinal efficiency to be recovered.

1.1 Related Literature

The interesting work of Bogomolnaia and Moulin (2001) has triggered a rapidly growing literature on ordinal efficiency and the PS mechanism:[6] Katta and Sethuraman (2006) extend PS to the full preference domain where indifference among alternatives is allowed using techniques from network flow theory. Kojima and Manea (2006) show that PS becomes fully strategy-proof when the market size gets sufficiently large. Manea (2006) shows that ordinal inefficiency of RP prevails even for large assignment problems. A similar (but logically unrelated) result to our main theorem is recently given by Che and Kojima (2008) who show that when sufficiently many copies of each object type are available, PS and RP are asymptotically equivalent.

In the literature various mechanisms that employ the top trading cycles method have been studied in contexts where resources are collectively owned. For the assignment problem, *essentially* the only Pareto efficient and strategy-proof mechanisms have been shown to be those that employ the top trading cycles method (e.g., see the references listed in footnote 3). An intuitive top trading cycles mechanism proposed for the context of *on-campus housing* by Abdulkadiroğlu and Sönmez (1999) has been theoretically as well as experimentally (e.g., see Chen and Sönmez, 2003) shown to have superior efficiency properties than a widely used real-life mechanism. For the *school choice problem* new and promising mechanisms that also operate on this method have been proposed in the recent literature as more efficient alternatives to real-life mechanisms (e.g., see Abdulkadiroğlu and Sönmez (2003b) and Kesten (2006a,b)). For the *kidney exchange problem* an inventory of mechanisms based on the same method have been proposed and advocated as attractive replacements for current real-life practises (e.g., see Roth, Sönmez, and Unver (2004) and, Sönmez and Unver (2006)).

Serial dictatorship mechanisms and all top trading cycles based allocation mechanisms mentioned above share one important common feature for the deterministic settings they have been studied: Pareto efficiency. In a random assignment context however, their corresponding counterparts RP and CfRE lack the appealing ordinal efficiency property. To the best of our knowledge, TTCfED and RP $[*k]$ are the first adaptations of these allocation methods to the random assignment context that can also achieve ordinal efficiency.

The paper is organized as follows: Section 2 introduces the formal model and gives the basic definitions. Section 3 describes the three central mechanisms. Section 4 presents the equivalence result, and Section 5 contains the analysis of the connection between ordinal efficiency, Pareto efficiency, and the top trading cycles idea. All the proofs are given in the Appendix.

[6] Cres and Moulin (2001) initially introduce PS for the case where each agent has the same ranking of objects. Bogomolnaia and Moulin (2002) offer a characterization of this mechanism for the same context.

2 The Model

Let $N \equiv \{1, 2, \ldots, n\}$ denote the finite set of agents. Let $H \equiv \{h_1, h_2, \ldots, h_n\}$ denote the finite set of objects. Each agent $i \in N$ is equipped with a complete, transitive, and antisymmetric preference relation $\succcurlyeq_i$ over H. Let $\succ_i$ denote the asymmetric part of $\succcurlyeq_i$. Let $\succcurlyeq \equiv (\succcurlyeq_i)_{i \in N}$.

A (deterministic) *assignment* is a bijection $\mu : N \to H$, and can be represented as a permutation matrix π (a $n \times n$ matrix with entries 0 or 1 and exactly one non-zero entry per row and one per column). Let $\mathcal{A}$ denote the set of all assignments. An assignment μ is *Pareto efficient* if there is no $\mu' \in \mathcal{A}$ such that $\mu'(i) \succcurlyeq_i \mu(i)$ for all $i \in N$, and $\mu'(i) \succ_i \mu(i)$ for some $i \in N$.

A *random allotment* is a probability distribution over H. Let ΔH denote the set of all random allotments. A *lottery* $\mathcal{L} = \sum_\mu \alpha_\mu \mu$ is a probability distribution over assignments where $\alpha_\mu \in [0, 1]$ and $\sum_\mu \alpha_\mu = 1$. A *random assignment* $P = [p_{ix}]_{i \in N,\, x \in H}$ is a bistochastic matrix where $p_{ix} \in [0, 1]$ denotes the probability that agent i receives object x. Let P_i be the random allotment of agent i at P. Note that

$$\forall i \in N \text{ and } \forall x \in H;\ p_{ix} \in [0, 1] \text{ and} \sum_{j \in N} p_{jx} = \sum_{y \in X} p_{iy} = 1.$$

Given an assignment μ represented by the permutation matrix $\pi(\mu)$, the lottery $\mathcal{L} = \sum_\mu \alpha_\mu \mu$ induces the random assignment $P = \sum_\mu \alpha_\mu \pi(\mu)$. Given any random assignment, by the well-known Birkhoff-von Neumann theorem there is at least one lottery that induces it. A *(random assignment) problem* is a preference profile $\succcurlyeq$. A *mechanism* φ is a function that associates with each problem $\succcurlyeq$ a random assignment $\varphi(\succcurlyeq)$.

2.1 Ordinal efficiency

A random assignment is *ex post* efficient if it can be represented as a probability distribution over Pareto efficient assignments. A popular ex post efficient mechanism is the *random priority (RP)* which is often used in real life:[7] A random ordering of agents is drawn from the uniform distribution; the first agent gets his favorite object, the second agent gets his favorite object among the remaining objects, and so on. Bogomolnaia and Moulin (2001) propose a more desirable and even stronger notion of efficiency than ex post efficiency.

Given a preference profile $\succcurlyeq$, and two random assignments P and Q, P *stochastically dominates* Q at $\succcurlyeq$ if and only if

$$\sum_{y \succcurlyeq_i x} p_{iy} \geq \sum_{y \succcurlyeq_i x} q_{iy} \ \forall i \in N, \forall x \in H.$$

[7] It must be noted however that RP is no longer ex post efficient once indifference in preferences is allowed.

A random assignment is *ordinally efficient* if it is not stochastically dominated by another random assignment.[8] Surprisingly, the outcome of RP may not always be ordinally efficient (Bogomolnaia and Moulin, 2001). If a lottery induces an ordinally inefficient random assignment, then such a lottery is not unconstrained efficient when agents are expected utility maximizers.

3 Three mechanisms

3.1 Probabilistic Serial

Bogomolnaia and Moulin (2001) provide an interesting characterization of the set of ordinally efficient random assignments for any given random assignment problem. Based on this result they propose a new mechanism, called the *probabilistic serial (PS)*, that always chooses the unique central point in this set. The PS outcome is calculated via the following algorithm:

The Probabilistic Serial (PS) Algorithm

Think of each object as an infinitely divisible good. Allotting p_{ix} units of object x to agent i simply means that agent i gets object x with probability p_{ix}. Each agent "eats" (representing his allotment of that object) away from his favorite object at the same speed until the object is completely exhausted, and continues with his next favorite object until no object remains. More precisely:

Step 1: Each agent eats away from his favorite object at the same speed. Stop when an object is completely exhausted.

Step k, $2 \leq k \leq n$: Consider the remaining objects with the remaining units of them. Each agent eats away from his remaining favorite object at the same speed. Stop when an object is completely exhausted.

The algorithm terminates in at most n steps. In fact, one can obtain the complete set of ordinally efficient random assignments simply by varying the speed at which an agent can eat away from his favorite object at a given time during the algorithm (Bogomolnaia and Moulin, 2001).

Bogomolnaia and Moulin (2001) argue that PS is the most natural fair selection from the ordinally efficient set. In addition to ordinal efficiency, its outcome is envy-free, and the mechanism is weakly strategy-proof. When there are at most three agents, the mechanism is uniquely characterized by the three requirements. We next give a simple example.

[8] See Abdulkadiroğlu and Sönmez (2003a) for an interesting characterization of ordinally efficient random assignments. Also see McLennan (2002) for a result concerning the relationship between ordinal efficiency and ex ante efficiency.

Example 1: Let $N = \{1, 2, 3, 4\}$ and $H = \{a, b, c, d\}$. Preferences are as follows:

$\succcurlyeq_1$: $a\ b\ c\ d$

$\succcurlyeq_3$: $b\ a\ c\ d$

$\succcurlyeq_2$: $a\ b\ d\ c$

$\succcurlyeq_4$: $b\ c\ d\ a$

Step 1: Agents 1 and 2 eat away from object a, while 3 and 4 eat away from object b. We stop when objects a and b get completely exhausted. At this point each one of agents 1 and 2 has eaten 1/2 units of object a, and each one of 3 and 4, 1/2 units of object b.

Step 2: Agents 1, 3, and 4 eat away from object c, while 2 eats away from object d. We stop when object c gets completely exhausted when each one of 1, 3, and 4 has eaten 1/3 units of it. At this point agent 2 has eaten 1/3 units of object d.

Step 3: All agents eat away from the 2/3 remaining units of object d. We stop when object d gets completely exhausted when each agent has eaten 2/12 units of it. The overall random assignment that the PS algorithm yields is given below. [In what follows we will use an $n \times n$ table to represent a random assignment matrix where the rows correspond to agents and the columns correspond to objects, i.e., a specific entry p_{ix} shows the probability that agent i receives object x.]

	a	b	c	d
1	1/2	0	1/3	1/6
2	1/2	0	0	1/2
3	0	1/2	1/3	1/6
4	0	1/2	1/3	1/6

3.2 Top Trading Cycles from Equal Division

An increasingly popular indivisible good allocation method is Gale's celebrated *top trading cycles (TTC)*. This procedure was first proposed in the context of a *housing market* (Shapley and Scarf, 1974). In such a market each agent is initially *assigned* (or, endowed with) a different object. The TTC method is applied to such a market as follows: Each agent points to the agent who is assigned his favorite object. Since the number of agents is finite, there is at least one cycle. Within each cycle the corresponding trades are carried out (i.e., each agent in a cycle is allotted the object that was assigned to the agent he is pointing to), and each agent in every cycle is removed. Next consider the new market, and apply the same method, and so on. This method yields the unique core allocation (Roth and Postlewaite, 1979) of the market which also coincides with the competitive equilibrium outcome (Shapley and Scarf, 1974).

Abdulkadiroğlu and Sönmez (1998) introduce a first adaptation of the TTC method to the random assignment problem: A random initial assignment of objects is drawn from the uniform distribution, and the TTC method is applied to the resulting housing market. They call it *core from random endowments (CfRE)*. Quite interestingly, RP and CfRE turn out to be equivalent (Abdulkadiroğlu and Sönmez, 1998).

In this paper we offer an alternative way to utilize the TTC idea in our context. But first consider the following interpretation of the problem: Initially all resources are collectively owned, and each agent a priori has equal right over each object. This suggests that each agent should initially be entitled to any object with probability $1/n$. This implies that a natural starting point is a situation where each agent is initially assigned probability $1/n$ of each object.

Next is an adaptation of the TTC method to our context, which builds upon the above interpretation. Two difficulties arise with such an approach: (1) in all earlier applications of the top trading cycles method, an agent trades an object assigned to him for another object assigned to some other agent, whereas here an agent may be assigned unequal *fractions* of several objects; (2) in all earlier applications an agent participates in at most one cycle, whereas here an agent may participate in more than one cycle, and can even be contained in multiple nested cycles. The first problem can intuitively be dealt with by allowing each agent to uniformly trade his assignment(s) along each cycle containing him. To ease the difficulty with the second problem, we only look for self-cycles and pairwise trading cycles (i.e., those consisting of two agents) thereby allowing only direct exchanges. (See Remark 4 for more on this.)

Hence our adaptation of TTC is as follows: We initially assign each agent a $1/n$ share of each object. Then we apply a simple version of the TTC method in which (1) agents are allowed to trade *equal* units of objects along *each* cycle, and (2) *only* single and pairwise cycles are considered. We call this mechanism *top trading cycles from equal division (TTCfED),* and its outcome is calculated via the following algorithm (where the above argument is made more precise):

The Top Trading Cycles Algorithm from Equal Division (TTCfED)

Think of each agent as consisting of n pseudo-agents each of whom is initially assigned $1/n$ units of a different object. An agent's random allotment of a particular object is the sum of all the units of that object he is allotted. Consider any agent i.

Step k, $1 \leq k \leq n$: Each pseudo-agent of agent i points to every pseudo-agent who is assigned a positive unit of agent i's favorite object. More precisely, if a pseudo-agent of agent i is already assigned a positive unit of agent i's favorite object, then he points to himself, and forms a self-cycle; otherwise he points to every pseudo-agent that is assigned a positive unit of his favorite object. There is at least one cycle. (Moreover, a pseudo-agent may even participate in multiple cycles.) Next the corresponding trades are performed: (i) *self-cycles: agent i is allotted all the units assigned to his pseudo-agent who forms the self-cycle, and that pseudo-agent is removed; (ii) pairwise trading cycles: within each cycle a pseudo-agent forms, he trades equal units of the object he is assigned for equal units of the object the pseudo-agent he is pointing to is assigned. This trade continues until the point there is a pseudo-agent who has no units (of the object he is assigned) left to trade. At this point all trades stop, and all pseudo-agents without any object to trade are removed.*

Remark 1: At any given step of the TTCfED algorithm, any two pseudo-agents who are assigned the same object *must* have equal units of that object. This is clearly true for the first step. Indeed, suppose it is also true for any step t with $1 < t < t'$ (induction hypothesis). Given any two distinct agents $i, i' \in N$ and some object $x \in H$, consider the two pseudo-agents (i, x) and (i', x) of step t that are not removed at the end of this step. This means object x is not the favorite object of either agent at step t (for otherwise the corresponding pseudo-agent would form a self-cycle and be removed at the end of step t). Then let $y, y' \in H \backslash \{x\}$ be the favorite objects of i and i' respectively at step t. If object x is not the favorite object of any agent in $N \backslash \{i, i'\}$ at step t, then neither of the pseudo-agents (i, x) and (i', x) forms a pairwise cycle at step t, and thus both pseudo-agents have equal units of x at step t'. Otherwise there is some agent $j \in N \backslash \{i, i'\}$ whose favorite object at step t is x. The induction hypothesis together with the fact that $x \succ_j y, y'$ implies that pseudo-agents (j, y) and (j, y') have not been removed before step t. Then pseudo-agents (i, x) and (j, y) form a trading cycle. Similarly, pseudo-agents (i', x) and (j, y') also form a trading cycle. Since the amount of trade within each trading cycle at a given step is the same, by the induction hypothesis, the two pseudo-agents have to have equal units of x at step t'.

Remark 2: The TTCfED algorithm is well-defined because a pairwise or a self-cycle always exists at each step. Indeed, consider any step $t \geq 1$ and any agent $i \in N$ whose favorite object at this step is some $x \in H$. If (i, x) has not been removed yet, then it forms a self-cycle. Otherwise, each pseudo-agent of i points to some pseudo-agent (j, x) with $j \in N \backslash \{i\}$. If x is the favorite object of agent j at this step, then pseudo-agent (j, x) forms a self-cycle. If some $x' \in H \backslash \{x\}$ is the favorite object of agent j at this step, then this means that $x \succ_i x'$. Since x' is not the favorite object of agent i at any step $t' \leq t$, by Remark 1 pseudo-agent (i, x') has not been removed before step t. Then pseudo-agents (i, x') and (j, x) form a pairwise cycle at step t.

Remark 3: The TTCfED algorithm terminates in at most n steps. This is because at least one object is removed at a given step of the procedure. Indeed, consider any step $t \geq 1$. Suppose that each agent $j \in N$ has the same favorite object $x \in H$. By Remark 1 any two pseudo-agents (j, x) and (j', x) with $j, j' \in N$ that are assigned object $x \in H$ have equal units of x. Then each pseudo-agent (j, x) forms a self-cycle, and object x is removed at the end of the step. Suppose that there are agents with different favorite objects.

Then by the argument in Remark 2, each $i \in N$ with some favorite object $x \in H$ forms a pairwise cycle with each $j \in N \backslash \{i\}$ with some favorite object $x' \in H \backslash \{x\}$ through pseudo-agents (i, x') and (j, x). By part (ii) of the algorithm, all pairwise trades stop at the point when some pseudo-agent (l, z) with $l \in N$ and $z \in H$, has no units of object z left to trade. Since the amount of trade within each trading cycle at a given step is the same, by Remark 1 every pseudo-agent (l', z) with $l' \in N$ participating in a pairwise cycle also has no units of object z left. On the other hand, any pseudo-agent (l'', z) with $l'' \in N$ who is not participating in a pairwise cycle is clearly forming a self-cycle. Hence, object z is removed at the end of step t.

Remark 4: Allowing for three-way or larger exchanges within the TTCfED algorithm does not introduce any new trading opportunities for any agent. In fact, any trade that can be identified at any step of the TTCfED algorithm involving $k > 2$ distinct agents is implemented through pairwise cycles by the above TTCfED algorithm. To see this point consider any step $t \geq 1$ of the algorithm. Let $I = \{i_1, i_2, \ldots, i_k\} \subset N$ be a set of distinct agents where each $i_j \in I$ has a distinct assignment $a_j \in H$. Suppose that the favorite object of each $i_j \in I$ at this step is assignment a_{j+1} of the next agent where $i_{k+1} \equiv i_1$. Clearly, such preferences give rise to the possibility of a $k-$way trade among agents in the set I (i.e., agent i_1 gets some unit of assignment a_2 of agent i_2; agent i_2 gets some unit of assignment a_3 of agent i_3;...; agent i_k gets some unit of assignment a_1 of agent i_1). Now consider the TTCfED algorithm. Since each agent in I has a distinct favorite object at this step, by the simple argument in Remark 2, for each agent $i_j \in I$, each remaining agent $i_l \in I \backslash \{i_j\}$ has a corresponding pseudo-agent (i_l, a_{j+1}) who is assigned the favorite object of agent i_j. This means that each agent $i_j \in I$ forms a pairwise cycle with each remaining agent $i_l \in I \backslash \{i_j\}$ to trade their favorite objects (e.g., agent i_1 gets some unit of assignment a_2 from each agent in the set $\{i_2, i_3, \ldots, i_k\}$).[9]

Example 1 (Cont'd): We now compute the outcome of TTCfED for the problem in Example 1. Initially each agent is assigned 1/4 units of each one of the four objects, and each agent i consists of 4 pseudo-agents: (i, a), (i, b), (i, c), and (i, d).

Step 1i: The favorite object of agents 1 and 2 is a, while that of 3 and 4 is b. Each agent's pseudo-agent points to each pseudo-agent who is assigned his favorite object (Figure 1). We first identify self-cycles: Each one of the pseudo-agents $(1, a)$, $(2, a)$, $(3, b)$, and $(4, b)$ forms a self-cycle, and is removed. Hence, each corresponding agent is allotted 1/4 units of his favorite object.

Step 1ii: Pairwise trading cycles: Each one of $(1, b)$ and $(2, b)$ forms a cycle with each one of $(3, a)$ and $(4, a)$. Recall that each pseudo-agent is assigned 1/4 units of some object. We next identify the cycle(s) in which

[9] Loosely speaking, the fact that objects are homogeneously distributed across agents at each step (Remark 1) makes large cycles irrelevant from an efficiency point of view. In a related indivisible good allocation problem of "kidney exchange," Unver, Roth, and Sönmez (2004) report that considering only pairwise and three-way cycles often suffices to obtain efficiency.

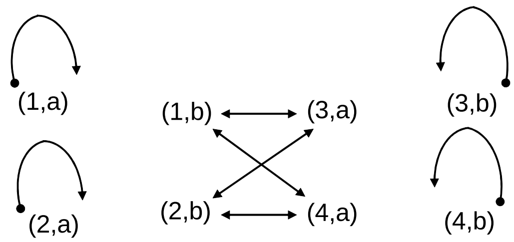

Figure 1: Step 1 of TTCfED for Example 1. [A one-sided arrow represents a self-cycle, and a two-sided arrow represents a pairwise cycle.]

trade ends first. Each of the four pseudo-agents participates in two cycles, and all trades end at the same time. Within each trading cycle of this step pseudo-agents exchange 1/8 units of their favorite objects for 1/8 units of the objects they are assigned. Hence, each one of 1 and 2 is allotted 2/8 units of a, and each one of 3 and 4 is allotted 2/8 units of b. All the pseudo-agents in these trading cycles are removed.

Step 2i: At the end of step 1 no pseudo-agent with any positive units of object a or b remains. Now the favorite object of 1, 3, and 4 is c, while that of 2 is d (Figure 2). We first identify self-cycles: Each one of $(1, c)$, $(3, c)$, and $(4, c)$ forms a self-cycle, and is removed. Each corresponding agent is allotted 1/4 units of c. Similarly, $(2, d)$ forms a self-cycle, and agent 2 is allotted 1/4 units of d.

Step 2ii: Pairwise trading cycles: Each one of $(1, d)$, $(3, d)$, and $(4, d)$ forms a cycle with $(2, c)$. Trade stops when within each trading cycle the corresponding pseudo-agents exchange 1/12 units of their favorite objects for 1/12 units of the objects they are assigned. Hence, each one of 1, 3, and 4 is allotted 1/12 units of c, and 2 is allotted 3/12 units of d. At this point each of $(1, d)$, $(3, d)$, and $(4, d)$ is left with 1/6 units of d. Pseudo-agent $(2, c)$ is removed.

Step 3: The only remaining pseudo-agents are $(1, d)$, $(3, d)$, and $(4, d)$, each with 1/6 units. Each one of 1, 3, and 4 is allotted 1/6 units of d through the corresponding self-cycles. The overall random assignment that the TTCfED algorithm yields is given below:

	a	b	c	d
1	1/2	0	1/3	1/6
2	1/2	0	0	1/2
3	0	1/2	1/3	1/6
4	0	1/2	1/3	1/6

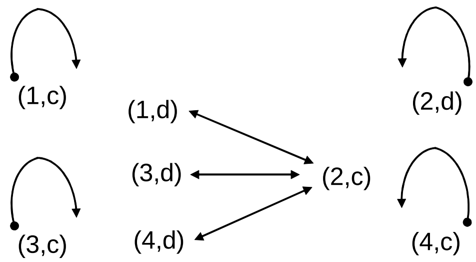

Figure 2: Step 2 of TTCfED for Example 1. [A one-sided arrow represents a self-cycle, and a two-sided arrow represents a pairwise cycle.]

3.3 Random Priority [$*k$]

The random priority (RP) mechanism when applied to the problem in Example 1 induces an ordinally inefficient assignment (see the random assignment below). A closer examination of this example shows why: The serial dictatorship (SD) associated with the ordering 1-2-3-4 of agents allots agent 2 object b (the favorite object of 3), whereas the SD associated with the ordering 4-3-1-2 of agents allots agent 3 object a (the favorite object of 2). But this causes efficiency loss in the induced random assignment because the two agents would both be better off (in the stochastic sense) if they were to trade their allotment probabilities of the two objects from the two SD applications. We argue that this phenomenon might be related to the fact that each particular SD restricts agents' choices to rapidly shrinking sets of resources. For example, when it is the turn of the r^{th} agent in the order where $r \in \{1, 2, \ldots, n\}$, he needs to pick his favorite choice among the remaining $(n - r + 1)$ objects. Since RP allows each agent to experience every possible turn in an ordering, situations (similar to above) in which a group of agents among themselves would rather trade their allotment probabilities across different orderings may easily arise, and such situations eventually lead to ordinal efficiency loss in the induced overall random assignment.

In order to remedy the above efficiency loss in RP we propose the idea of introducing artificial copies of objects into a problem as a way to give agents more flexibility over object choices. More precisely, we consider the following intuitive relaxation of the RP procedure: For any given ordering f of agents, to obtain the outcome induced by f, suppose that we now consecutively apply SD^f to a new problem obtained by introducing two extra copies of each object. Suppose for Example 1 that we were to introduce two extra copies of each object (leading to three copies of each object overall), and apply SD^f three times consecutively until all objects are allotted. For instance, when f is 1-2-3-4, these three consecutive applications yield the following three "allocations" in turn:[10]

$$\alpha_1 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ a & a & b & b \end{pmatrix},\ \alpha_2 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ a & b & c & c \end{pmatrix},\ \text{and } \alpha_3 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ c & d & d & d \end{pmatrix}.$$

Suppose further that we were to do the same exercise for each of the 24 possible orderings, and now take the average of $3 * 24$ SD outcomes to find an overall random assignment. The resulting random assignment is given below: [Also given is the outcome of this procedure when one introduces only one extra copy of each object.]

	a	b	c	d
1	11/24	1/12	1/4	5/24
2	11/24	1/12	0	11/24
3	1/12	5/12	1/3	1/6
4	0	5/12	5/12	1/6

RP

	a	b	c	d
1	1/2	0	1/3	1/6
2	1/2	0	0	1/2
3	0	1/2	1/3	1/6
4	0	1/2	1/3	1/6

RP [$*2$]

	a	b	c	d
1	35/72	2/72	23/72	12/72
2	35/72	2/72	0	35/72
3	2/72	34/72	24/72	12/72
4	0	34/72	25/72	13/72

RP [$*3$]

We call the mechanism whose outcome is the random assignment that is induced as the average of $kn!$ serial dictatorship outcomes as the *random priority (RP)* [$*k$]. When computing RP [$*k$], for every possible ordering of agents the associated SD is applied consecutively k times to a $k-$replica of resources until all objects are exhausted. Clearly, when $k = 1$, this mechanism is equivalent to RP.

4 Main Result

Surprisingly, the outcomes of PS and TTCfED are the same for the random assignment problem given in Example 1.[11] On the other hand, the ordinally efficient RP [$*2$] assignment given above seems to entail hints that might shed light to questions about the efficiency loss of the RP mechanism. Our main result establishes a three-way link between the three different classes of mechanisms.

[10] Here we use the term *allocation* as an extension of an *assignment* to the case when multiple copies of objects are allowed. More precisely, given N and H let c_x denote the number of copies of object $x \in X$. Then an allocation is a mapping $\alpha : N \to H$ such that $\alpha(i) \in H$ and, for each $x \in X$, $|\alpha^{-1}(x)| \leq c_x$.

[11] For PS and TTCfED mechanisms however, the outcomes are not the same *stepwise,* i.e., an agent's allotment at some step of the PS algorithm may be different than his allotment at the same step of the TTCfED algorithm. For instance, in Example 1 at the end of step 2, PS allots 1/3 units of object d to agent 2, whereas TTCfED allots him 1/2 units of this object. Also, under PS, the amount of object any two agents are allotted at a given step is the same, which again is not true under TTCfED.

Theorem 1*: The following mechanisms are equivalent:*[12]

(i) Probabilistic serial

(ii) Top trading cycles from equal division

(iii) $\lim_{k\to\infty}$ *Random priority* $[*k]$

The above equivalences all hint at a common intuition: In random assignment problems mechanisms tend to gain efficiency as they better utilize the richer probabilistic setup so as to mitigate any mishaps arising from the 'indivisibility' of resources. An immediate implication of Theorem 1 is that both TTCfED and $\lim_{k\to\infty} RP$ $[*k]$ are ordinally efficient. Consequently, from an efficiency standpoint TTCfED improves upon CfRE by initially dividing each object into n equal shares whereas RP $[*k]$ improves upon RP when each object is reproduced infinitely many times.

5 Conclusion

Random priority (RP) and probabilistic serial (PS) are two important competitors for the random assignment problem. The first part of Theorem 1 (i.e., (i)⇔(ii)) together with the equivalence result of Abdulkadiroğlu and Sönmez (1998) suggests that both RP and PS are essentially top trading cycles mechanisms, and that the main difference between them comes from the way they choose the 'initial conditions of trade:' While RP initially assigns each object to each agent with equal probability (i.e., probability $1/n!$) before allowing them to trade, PS initially assigns an equal probability (i.e., probability $1/n$) of each object to each agent before allowing them to trade. The latter initial conditions give the efficiency advantage to PS by providing a sufficiently thick market in terms of trading opportunities. Furthermore, the fact that TTCfED is ordinally efficient also highlights the power of the top trading cycles procedure for indivisible goods allocation by showing that even a simple version of this idea (with self and pairwise cycles) can be sufficient to achieve a strong form of efficiency.

We close this section with a final remark on RP $[*k]$. Recall our earlier discussion about the (ordinal) inefficiency of RP. To follow up, in proving the second part of Theorem 1 (i.e., (i)⇔(iii)) we show that when k becomes sufficiently large, the variation across induced individual serial dictatorship outcomes becomes negligibly small. This finding supports our earlier intuition about the underlying reason behind the ordinal inefficiency of RP. In fact, it can even be shown for arbitrary k that if every serial dictatorship ordering of agents induces the same $k-$allocations for a problem with a $k-$replica of the initial resources, then the RP $[*k]$ outcome coincides with the PS outcome, and thus becomes ordinally efficient.

[12] Two mechanisms Φ and Ψ are equivalent if and only if for any problem $\succcurlyeq$, $\Phi(\succcurlyeq) = \Psi(\succcurlyeq)$.

6 The Appendix

Proof of Theorem 1: We first show (i)=(ii) and next that (i)=(iii). Fix a problem $\succcurlyeq$.

Proof of (i)=(ii):

Step 1: Comparison of the two algorithms. We first claim that whichever favorite object an agent has at any given step of the PS algorithm, he has the same favorite object at the same step of the TTCfED algorithm. For a given step t with $1 \leq t \leq n$, let H^t_{PS} denote the set of objects which are still available at the end of step t of the PS algorithm, and H^t_{TTC} the set of objects which are still available at the end of step t of the TTCfED algorithm. Then it is sufficient to show that $H^t_{PS} = H^t_{TTC}$ for all t with $1 \leq t \leq n$.

Let us denote by k^t_x the number of agents each of whom has x as his favorite object at step t of the PS algorithm, and by l^t_x the same number corresponding to step t of the TTCfED algorithm. Let $k^t = \max_{x \in H} k^t_x$ with $1 \leq t \leq n$. Similarly, let $l^t = \max_{x \in H} l^t_x$ with $1 \leq t \leq n$.

We prove our claim by induction. Let $t = 1$. Clearly, $k^1_x = l^1_x$ for all $x \in H$. If $k^1 = l^1 = n$, then this means under the PS algorithm all n agents eat away from the same object at step 1, and each gets $1/n$ units of it. For the same case, under the TTCfED algorithm each agent forms a self-cycle for the same object, and again gets $1/n$ units of it. If $k^1 = l^1 \neq n$, then for the PS algorithm, any object that gets exhausted at step 1 solves:

$$\min_{x \in H} \frac{1}{k^1_x} \tag{1}$$

For the TTCfED algorithm, the object whose trade ends first at step 1 solves:

$$\min_{x \in H} \frac{1}{n l^1_x} \tag{2}$$

At the end of step 1 of the TTCfED algorithm, any agent whose favorite object is different than a minimizer of (2) has no units of that minimizer left. Any agent whose favorite object is a minimizer of (2) also has no units of that minimizer left since he is allotted all $1/n$ units of it through the self-cycle he forms. Hence, there can not remain any units of any minimizer of (2) at the end of step 1. All other objects are still available (possibly with fewer units). Then, since any solution to (1) is also a solution to (2), we have $H^1_{PS} = H^1_{TTC}$.

Now suppose that for any step $1 < t < t'$, we have $H^t_{PS} = H^t_{TTC}$. This implies that $k^t = l^t$ for all $t \leq t'$. Let $EPA^t > 0$ denote the amount of object eaten away by any agent at step t with $1 \leq t \leq n$, of the PS algorithm. Let $TPC^t \geq 0$ denote the amount of object traded in any cycle at step t with $1 \leq t \leq n$, of the TTCfED algorithm. We consider two cases:

Case 1. *For all* $t \leq t'$, $k^t = l^t \neq n$ (i.e., at each step there are at least two agents with different favorite objects): We claim that $EPA^t = nTPC^t$ for any $t \leq t'$. Note that $EPA^1 = 1/k^1$ and $TPC^1 = 1/nl^1$. So the claim clearly holds for $t = 1$. The amount of object that gets eaten away by any agent at step $t \leq t'$ of the PS algorithm is determined by:

$$EPA^t = \min_{x \in H_{PS}^{t-1}} \frac{1 - k_x^1 EPA^1 - \cdots - k_x^{t-1} EPA^{t-1}}{k_x^t} \tag{3}$$

where the term in the numerator is the amount of object x available at the beginning of step t, and k_x^t is the number of agents who eat away from object x at step t. Any object that gets exhausted at the end of step t is a minimizer of the problem in (3). All other objects in H_{PS}^{t-1} are still available (possibly with fewer units as compared to the previous step). Similarly, the amount of object that is traded in any cycle at step $t \leq t'$ of the TTCfED algorithm is determined by:

$$TPC^t = \min_{x \in H_{TTC}^{t-1}} \frac{1/n - l_x^1 TPC^1 - \cdots - l_x^{t-1} TPC^{t-1}}{l_x^t} \tag{4}$$

where the term in the numerator (by Remark 1) is the amount of object x assigned to each one of those pseudo-agents who still have positive units of x at the beginning of step t (more precisely: each such agent is initially assigned $1/n$ units of object x, and at each step s he forms l_x^s trading cycles in each one of which he trades TPC^s units of x for his favorite object available at that step) and k_x^t is the number of agents who form trading cycles to trade object x at step t.

Since for any step $t < t'$, $H_{PS}^t = H_{TTC}^t$ by the induction hypothesis, we have $k_x^t = l_x^t$ for any object $x \in H_{PS}^{t-1} = H_{TTC}^{t-1}$ and any step $t \leq t'$. Then since $EPA^1 = nTPC^1$, and since any minimizer of the problem in (3) is also a minimizer of the problem in (4) for any step $t \leq t'$, it follows that $EPA^t = nTPC^t$ for any $t \leq t'$. For a given step $t \leq t'$, let x_t^* be a solution to the problems in (3) and (4). Using a similar argument as before, there can not remain any units of x_t^* at the end of step t, and all other objects in H_{TTC}^{t-1} are still available (possibly with fewer units). Thus, we conclude that $H_{PS}^{t'} = H_{TTC}^{t'}$.

Case 2. *There is* $s \leq t'$ *such that* $k^s = l^s = n$: Let $s \leq t'$ be the earliest step with $k^s = l^s = n$. Let x_s^* be the favorite object of every agent at step s of both algorithms. Under both algorithms, at the end of step s object x_s^* gets exhausted, and there is no change in the available units of all other objects. More precisely, since for any $x \in H$, both k_x^t and l_x^t are non-decreasing in t as long as object x is still available, at the end of step s there has to be exactly a total of one unit available from each remaining object . Thus,

$H^s_{PS} = H^s_{TTC}$. Now we have a smaller version of the initial problem, and for any remaining step until step t', we can iteratively re-visit cases 1 and 2 to finally conclude that $H^{t'}_{PS} = H^{t'}_{TTC}$.

Step 2: Establishing the equivalence. By step 1 the objects (not necessarily the units) that are available at every step of both algorithms are the same. Therefore each agent has the same favorite object at any given step of both algorithms. We show that the two mechanisms select the same random assignment. Take any agent $i \in N$ and any object $x \in H$. Suppose agent i eats away from object x from step t to step t', $t' \geq t$ of the PS algorithm. By step 1 this means object x is his favorite object from step t to step t' of the TTCfED algorithm as well. Note first that for any step s such that $t \leq s < t'$, we have $k^s_x \neq n$ because otherwise object x would be completely exhausted at the end of step s. We consider two cases:

Case 1. $k^{t'}_x \neq n$: Agent i's random allotment of object x under the PS mechanism is given by:

$$PS_i(\succcurlyeq)[x] = \sum_{s=t}^{t'} EPA^s \tag{5}$$

Agent i's random allotment of object x under the TTCfED mechanism is given by:

$$TTC_i(\succcurlyeq)[x] = (\frac{1}{n} - \sum_{s=1}^{t-1} k^s_x TPC^s) + \sum_{s=t}^{t'} (n - k^s_x) TPC^s \tag{6}$$

where the first term is the amount of x agent i is allotted through the self-cycle at step t (his initial assignment of x minus the amount of x he trades with other agents until step t), and the second term is the amount of x agent i is allotted through the trading cycles from step t to step t' (at any step s with $t \leq s \leq t'$, agent i forms a trading cycle with each one of those agents who have a different favorite object at step s, making a total of $n - k^s_x$ trading cycles). Note that by case 1 of step 1, for any s with $t \leq s \leq t'$, we have $EPA^s = nTPC^s$. Then rearranging (6), we have:

$$TTC_i(\succcurlyeq)[x] = (\frac{1}{n} - \sum_{s=1}^{t'} k^s_x TPC^s) + \sum_{s=t}^{t'} EPA^s \tag{6'}$$

Since object x gets completely exhausted at the end of step t', there is no agent with a positive amount of x at the end of step t'. This means the first term on the RHS of (6') is zero establishing the equivalence of (5) and (6).

Case 2. $k^{t'}_x = n$: Agent i's random allotment of object x under the PS mechanism is again given by (5). Agent i's random allotment of object x under the TTCfED mechanism is now given by:

$$TTC_i(\succcurlyeq)[x] = (\frac{1}{n} - \sum_{s=1}^{t-1} k_x^s TPC^s) + \sum_{s=t}^{t'-1} (n - k_x^s) TPC^s \tag{7}$$

where we now do not include any terms into (7) for step t' since agent i does not participate in any trading cycles at this step (the only cycles at this step are the self-cycles formed by those agents whose favorite object is x only at this step of the algorithm.).

The amount of object x allotted to agent i at step t' of the PS algorithm is given by:

$$EPA^{t'} = \frac{1 - \sum_{s=1}^{t'-1} k_x^s EPA^s}{n} \tag{8}$$

Inserting (8) in (5) and rearranging, we have:

$$PS_i(\succcurlyeq)[x] = (\frac{1}{n} - \frac{1}{n} \sum_{s=1}^{t'-1} k_x^s EPA^s) + \sum_{s=t}^{t'-1} EPA^s \tag{9}$$

Using $EPA^s = nTPC^s$ for all s with $t \leq s < t'$ by case 1 of step 1, we have

$$PS_i(\succcurlyeq)[x] = (\frac{1}{n} - \frac{1}{n} \sum_{s=1}^{t-1} k_x^s EPA^s) + \sum_{s=t}^{t'-1} (n - k_x^s) TPC^s \tag{9'}$$

Finally, for any s with $1 \leq s \leq t-1$, we have either $EPA^s = nTPC^s$ (again by case 1 of step 1), or $TPC^s = 0$. If $TPC^s = 0$ for some s, with $1 \leq s \leq t-1$, then $k_x^s = 0$. Then for any s with $1 \leq s \leq t-1$, $k_x^s EPA^s = nk_x^s TPC^s$. Inserting this into (9'), we get the equivalence with (7).

Proof of (i)=(iii): (Assume that the notation used in the above proof still applies.) It is easy to see that if $n < 3$, then PS is equivalent to RP $[*k]$ for all k. So suppose $n \geq 3$. Let k be sufficiently large so that $k \gg 2^{n+1} n^{2n-1}$. Let H_t denote the set of objects that are exhausted at step t of the PS algorithm. Thus, $\{H_t\}_{t=1}^n$ is a partition of H. Clearly, any $x_1 \in H_1$ solves the problem in (3) for $t = 1$. Hence, $EPA^1 = 1/k^1$.

Now consider RP $[*k]$. Fix an ordering f of agents. The first application of SD^f to problem $\succcurlyeq$ assigns each agent his favorite object at step 1 of the PS algorithm. Moreover, the outcome of SD^f is the exactly same allocation for the first $a_1 \equiv NSD_k^1 \equiv \lfloor k/k^1 \rfloor = \lfloor kEPA^1 \rfloor$ applications.[13] Let $Rem_k^s(x)$ denote the number of remaining copies of object $x \in H$ at the end of application s of SD^f. Note that for all $x_1 \in H_1$, we have $0 \leq Rem_k^{a_1}(x_1) < k^1$. This means that for all $x_1^* \in H_1$, $Rem_k^{a_1+1}(x_1^*) = 0$. On the other hand, for all $x \in H \backslash H_1$, $Rem_k^{a_1}(x) = k - k_x^1 NSD_k^1$. Since the number of copies of any object assigned at any application

[13] For any real number r, $\lfloor r \rfloor$ denotes the largest integer that is not strictly larger than r.

of SD^f cannot exceed n, for all $x \in H \backslash H_1$ we have:

$$Rem_k^{a_1+1}(x) - Rem_k^{a_1}(x) \leq n \tag{10}$$

Since for all $x_1 \in H_1$, $Rem_k^{a_1+1}(x_1) = 0$, application $(a_1 + 2)$ of SD^f assigns each agent his favorite object at step 2 of the PS algorithm. Note that by the choice of k, we have $Rem_k^{a_1}(x) \gg n$ for all $x \in H \backslash H_1$. Then since each $x_2 \in H_2$ solves the problem in (3) for $t = 2$, there exists some $x_2^* \in H_2$ that solves $NSD_k^2 \equiv \min_{x \in H \backslash H_1} \lfloor Rem_k^{a_1+1}(x)/k_x^2 \rfloor$. This means that applications $(a_1 + 2)$ through $a_2 \equiv a_1 + 1 + NSD_k^2$ of SD^f each result in the same allocation (i.e., the one in which each agent receives his favorite object at step 2 of the PS algorithm). Since each $x_2 \in H_2$ solves the problem in (3) for $t = 2$, inequality (10) implies that $0 \leq Rem_k^{a_2}(x_2) \leq n$ for all $x_2 \in H_2$.[14] On the other hand, by our choice of k, for all $x \in H \backslash (H_1 \cup H_2)$ we have $Rem_k^{a_2}(x) = Rem_k^{a_1+1}(x) - k_x^2 NSD_k^2 \gg n$. Then all $x_2 \in H_2$ is exhausted in at most n consecutive applications of SD^f after application a_2 of SD^f, i.e., for all $x_2 \in H_2$, $Rem_k^{a_2+n}(x_2) = 0$. This, in turn, means that application $(a_2 + n + 1)$ of SD^f assigns each agent his favorite object at step 3 of the PS algorithm.

We iteratively apply the above argument to obtain the number of applications of SD^f that assigns each agent his favorite choice at some step of the PS algorithm. Let a_t denote the number of applications of SD^f (beginning from the first application) until we obtain the last allocation in which each agent receives his favorite object at step t of the PS algorithm. Let NSD_k^t denote the number of applications of SD^f that yields the allocation in which each agent receives his favorite object at step t of the PS algorithm. Finally, let ξ_t denote the number of applications of SD^f as of application a_t until we obtain an allocation in which each agent receives his favorite object at step $t+1$ of the PS algorithm. Note that we have already computed that $\xi_1 \leq 1$ and that $\xi_2 \leq n$. In general, for all $t \geq 3$ and all $x \in H \backslash (\cup_{j=1}^{t-1} H_j)$ we have:

$$k - \sum_{s=1}^{t-1} k_x^s NSD_k^s \geq Rem_k^{a_{t-1}}(x) \geq k - \sum_{s=1}^{t-1} k_x^s NSD_k^s - n \sum_{s=1}^{t-2} \xi_s \tag{11}$$

By our choice of k, for all $x \in H \backslash (\cup_{j=1}^{t-1} H_j)$ and all $t \geq 3$, we have $Rem_k^{a_{t-1}}(x) \gg n \sum_{s=1}^{t-2} \xi_s$. Then since each $x_t \in H_t$ solves the problem in (3) for each step $t \geq 2$ of the PS algorithm, there exists some $x_t^* \in H_t$ that

[14] Formally, for all $x_2, x_2' \in H_2$, $|\frac{Rem_k^{a_1}(x_2)}{k_{x_2}^2} - \frac{Rem_k^{a_1}(x_2')}{k_{x_2'}^2}| = |\frac{k - k_{x_2}^1 \lfloor kEPA^1 \rfloor}{k_{x_2}^2} - \frac{k - k_{x_2'}^1 \lfloor kEPA^1 \rfloor}{k_{x_2'}^2}| = |\frac{k_{x_2}^1 kEPA^1 - k_{x_2}^1 \lfloor kEPA^1 \rfloor}{k_{x_2}^2} - \frac{k_{x_2'}^1 kEPA^1 - k_{x_2'}^1 \lfloor kEPA^1 \rfloor}{k_{x_2'}^2}| = |(kEPA^1 - \lfloor kEPA^1 \rfloor)(\frac{k_{x_2}^1}{k_{x_2}^2} - \frac{k_{x_2'}^1}{k_{x_2'}^2})| < 1$ where the second equality follows from the fact that $\frac{k - kk_{x_2}^1 EPA^1}{k_{x_2}^2} = \frac{k - kk_{x_2'}^1 EPA^1}{k_{x_2'}^2}$ which is implied by $x_2, x_2' \in H_2$. Then inequality (10) implies that $|\frac{Rem_k^{a_1+1}(x_2)}{k_{x_2}^2} - \frac{Rem_k^{a_1+1}(x_2')}{k_{x_2'}^2}| \leq n$. This, in turn, implies that $0 \leq Rem_k^{a_2}(x_2) \leq n$ for all $x_2 \in H_2$.

gets first exhausted among objects in $H\backslash(\cup_{j=1}^{t-1}H_j)$ during consecutive applications of SD^f, and it solves:

$$NSD_k^t \equiv \min_{x\in H\backslash(\cup_{j=1}^{t-1}H_j)} \lfloor \frac{Rem_k^{a_{t-1}+\xi_{t-1}}(x)}{k_x^t} \rfloor \tag{12}$$

This means that applications $(a_{t-1}+\xi_{t-1}+1)$ through $a_t \equiv a_{t-1}+\xi_{t-1}+NSD_k^t$ of SD^f each result in the allocation in which each agent receives his favorite object at step $t \geq 2$ of the PS algorithm. Since each $x_t \in H_t$ solves the problem in (3), inequality (11) implies that $0 \leq Rem_k^{a_t}(x_t) \leq n\sum_{s=1}^{t-1}\xi_s$ for all $x_t \in H_t$. Since the number of copies of any object assigned at any application of SD^f cannot exceed n, we must have $0 \leq \xi_t \leq n\Sigma_{s=1}^{t-1}\xi_s$ for all $t \geq 2$. On the other hand, by our choice of k, for all $x \in H\backslash(\cup_{j=1}^{t}H_j)$ with $2 \leq t \leq n-1$, we have $Rem_k^{a_t}(x) = Rem_k^{a_{t-1}+\xi_{t-1}}(x) - k_x^t NSD_k^t \gg n\sum_{s=1}^{t-1}\xi_s$.[15] Then each $x_t \in H_t$ with $t \geq 2$ is exhausted in $\xi_t \leq n\Sigma_{s=1}^{t-1}\xi_s$ consecutive applications of SD^f after application a_t of SD^f, i.e., for all $x_t \in H_t$, $Rem_k^{a_t+\xi_t}(x_t) = 0$.

Take any agent $i \in N$ and any object $x \in H$. If agent i eats away from object x from step t through step t' of the PS algorithm, then SD^f assigns him object x at applications $a_{t-1}+\xi_{t-1}+1$ through $a_{t'}$. Recall that $\xi_1 \leq 1$ and $\xi_t \leq n\Sigma_{s=1}^{t-1}\xi_s$ for all $t \geq 2$. Then since n is finite, $\Sigma_{s=1}^{n}\xi_s$ remains finite and independent of k as $k \to \infty$. Then because f is arbitrary, we can write:

$$\lim_{k\to\infty} RP_i[*k](\succcurlyeq)[x] = \lim_{k\to\infty} \frac{1}{k}\sum_{s=t}^{t'} NSD_k^s \tag{13}$$

When $t=1$, it is easy to see that $NSD_k^1/k \to 1/k^1 = EPA^1$ as $k \to \infty$. When $t \geq 2$, combining (11) and (12) applying the well-known Sandwich theorem (recall that n is finite), we get $NSD_k^t/k \to EPA^t$ (given by (3)) as $k \to \infty$. This means that RHS of (13) coincides with that of (5) completing the proof.

Q.E.D.

References

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

2. Abdulkadiroğlu, A. and T. Sönmez, House allocation with existing tenants, *Journal of Economic Theory* 88 (1999), 233-260.

[15] Given that $\xi_1 \leq 1$ and $\xi_t \leq n\Sigma_{s=1}^{t-1}\xi_s$, one can compute that $n\Sigma_{s=1}^{n}\xi_s < A \equiv (2n)^n$. We also need to choose k sufficently large so that $Rem_k^{a_t}(x) \gg n\sum_{s=1}^{t-1}\xi_s$ for all t with $n-1 \geq t \geq 2$. Since there are at most n steps of the PS algorithm, we have $Rem_k^{a_{n-1}}(x) > k/n^{n-1} - A$. Thus it suffices to choose $k \gg 2^{n+1}n^{2n-1}$.

3. Abdulkadiroğlu, A. and T. Sönmez, Ordinal efficiency and dominated sets of assignments, *Journal of Economic Theory* **112** (2003a), 157-172.

4. Abdulkadiroğlu, A. and T. Sönmez, School choice: A mechanism design approach, *American Economic Review.* **93** (2003b), 729-747.

5. Bogomolnaia, A. and H. Moulin, A new solution to the random assignment problem, *Journal of Economic Theory* **100** (2001), 295-328.

6. Bogomolnaia, A. and H. Moulin, A simple random assignment problem with a unique solution, *Economic Theory*, **19** (2002), 3, 623-636.

7. Che, Y.-K. and F. Kojima, Asymptotic equivalence of probabilistic serial and random priority mechanisms, mimeo (2008), Columbia University and Stanford University.

8. Chen, Y. and T. Sönmez, Improving efficiency of On-Campus Housing: An Experimental Study, *American Economic Review,* **92** (2002), 1669-1686.

9. Cres, H. and H. Moulin, Scheduling with opting out: improving upon random priority, *Operational Research.* **49** (2001), 565-577.

10. Ehlers, L., Coalitional strategy-proof house allocation, *Journal of Economic Theory* **105** (2002), 298-317

11. Hylland, A. and R. Zeckhauser, The efficient allocation of individuals to positions, *Journal of Political Economy* **91** (1979), 293-314.

12. Kagel, J. and A. Roth, eds., *The Handbook of Experimental Economics* (1995), Princeton University Press.

13. Katta, A.-K. and J. Sethuraman, A solution to the random assignment problem on the full preference domain (2006), *Journal of Economic Theory* **131** (2006), 231-250.

14. Kesten, O., On Two Competing Mechanisms for Priority Based Allocation Problems, *Journal of Economic Theory* **127** (2006a), 155-171.

15. Kesten, O., An inventory of top trading cycles mechanisms for school choice problems, mimeo (2006b), Carnegie Mellon University.

16. Kojima, F. and M. Manea Strategy-proofness of the probabilistic serial mechanism in large random assignment, mimeo (2006), Harvard University.

17. Manea, M., Asymptotic ordinal inefficiency of random serial dictatorship, *Theoretical Economics* forthcoming.

18. McLennan A., Ordinal efficiency and the polyhedral separating hyperplane theorem, *Journal of Economic Theory* **105** (2002), 435-449.

19. Pápai, S., Strategy-proof assignment by hierarchical exchange, *Econometrica* **68** (2000), 1403-1433.

20. Roth, A. and A. Postlewaite, Weak versus strong domination in a market with indivisible goods, *Journal of. Mathematical Economics* **4** (1977), 131-137.

21. Roth, A., T. Sönmez, and U. Ünver, Kidney Exchange, *Quarterly Journal of Economics* **19**, no. 2 (May 2004): 457-488.

22. Sönmez, T. and U. Ünver,.Kidney exchange with good samaritan donors: A characterization, mimeo (2006), Boston College.

23. Shapley, L. and H. Scarf, On cores and indivisibility, *Journal of Mathematical Economics* **1** (1974), 23-28.

24. Svensson, L.-G., Strategy-proof allocation of indivisible goods, *Social Choice and Welfare* **16** (1999), 557-567.

25. Zhou L., On a conjecture by Gale about one-sided matching problems, *Journal of Economic Theory* **52** (1990), 123-135.