Random Assignments of Bundles[*]

Shurojit Chatterji[†] Peng Liu[†]

September 12, 2018

Abstract

We study the random assignments of bundles with no free disposal. The key difference between the setting with bundles and the setting with objects (see Bogomolnaia and Moulin (2001)) is one of feasibility. The implications of this difference are significant. First, the characterization of sd-efficient random assignments is fundamentally different. Second, a possibility result in the setting with objects fails in the setting with bundles. However, in the setting with bundles, we are able to identify a preference restriction, called essential monotonicity, under which the random serial dictatorship rule (extended to the setting with bundles) is equivalent to the probabilistic serial rule (extended to the setting with bundles). This equivalence implies the existence of a rule on this restricted domain satisfying sd-efficiency, sd-strategy-proofness, and equal treatment of equals. Moreover, this rule selects only random assignments which can be decomposed as convex combinations of deterministic assignments.

Keywords: Random assignments; bundles; decomposability; sd-efficiency; sd-strategy-proofness; equal treatment of equals

JEL Classification: C78, D47, D71.

1 Introduction

We study the problem of allocating a finite set of objects to a finite set of agents, where money transfers are prohibited and each agent gets a bundle of objects. Each object has a certain number of identical copies. We refer to this number as the capacity of the object and require that it be smaller than the number of agents. A bundle is a subset of these objects. In particular, a bundle can take at most one copy of each object. In order to restore fairness, we adopt randomization in allocations.

A central assumption in earlier studies is that each agent gets at most one object (see for example Abdulkadiroğlu and Sönmez (1998) and Bogomolnaia and Moulin (2001)). However,

[*]This study was supported by the Ministry of Education, Singapore under the grant MOE2016-T2-1-168. We would like to thank Huaxia Zeng and Jingyi Xue for constructive suggestions. For valuable discussions, we would like to thank the participants of the SJET workshop, 2018.

[†]School of Economics, Singapore Management University, Singapore.

for many relevant applications, it is more appropriate to allocate the objects in bundles for at least two reasons. First, complementarity may require allocation in bundles in order to improve efficiency. Second, the total number of objects may be more than the number of agents and free disposal may not be acceptable. An example displaying these attributes is presented in Section 1.1.

In this setting, a random assignment rule is a function which takes as input preference profiles on bundles and selects a profile of lotteries on bundles, one for each agent. We investigate the existence of a random assignment rule that satisfies several desirable axioms.

The first axiom we consider is decomposability, which addresses the issue of implementing the selected random assignment. Put otherwise, the random assignment needs to be decomposed as a lottery over deterministically feasible assignments. Unfortunately, due to the combinatorial feature of the problem, a random assignment is in general not guaranteed to be decomposable. The second axiom deals with fairness and requires that whenever two agents report the same preference, they get the same lottery. To define the remaining axioms, we adopt the stochastic dominance extension, which states that a lottery is weakly better than another lottery if the former first-order stochastically dominates the latter.[1] The third axiom, sd-efficiency, requires that the rule selects only random assignments that are not Pareto dominated. The last axiom, sd-strategy-proofness, deals with incentive compatibility and requires that no unilateral deviation leads to a strictly better lottery.

Our first result (Theorem 1) characterizes sd-efficiency. More specifically, a random assignment is sd-efficient at a given preference profile if and only if it is unbalanced, which translates to the impossibility of identifying a Pareto improvement in a particular way. Moreover, we extend the simultaneous eating algorithm studied by Bogomolnaia and Moulin (2001) to the current setting and show that all random assignments generated by them are sd-efficient. However, there exists an sd-efficient random assignment at some profile that can not be generated by the simultaneous eating algorithm with any eating speed (see Remark 4).

We present subsequently an existence result. On a particular domain of preferences, the domain of essentially monotonic preferences, we show that there exists a decomposable random assignment rule satisfying sd-strategy-proofness, sd-efficiency, and equal treatment of equals. The set of essentially monotonic preferences varies with the capacities of objects. We first identify a subset of bundles as the critical bundles. Then essential monotonicity requires that a bundle contained in a critical bundle is less preferred to this critical bundle. Critical bundles are identified by letting agents sequentially take bundles and assuming that, whenever an agent is called upon, she takes one copy of each object that is still available. For example, if each object has 10 copies, each of the first 10 agent takes the whole bundle and the others take the empty bundle. Hence the set of critical bundles in this case contains only the whole bundle and the empty bundle. For another, if each of a and b has 10 copies and each of c and d has 5 copies, then each of the first 5 agents takes the whole bundle $abcd$; each of the next 5 agents takes ab; and all the others take the empty bundle. In this case, the set of critical bundles contains three bundles: $abcd$, ab, and the empty bundle. Notice that the more the capacities vary, the greater

[1]Such an extension is equivalent to that the former lottery generates an expected utility that is weakly higher than that delivered by the later, with respect to every Bernoulli utility representing the given ordinal preference.

the number of critical bundles that impose restrictions on preferences, and consequently the smaller the domain of essentially monotonic preferences.

According to the way we identify the critical bundles, the notion of essential monotonicity can be seen as modeling situations where all objects are "goods," so that whenever an agent is called upon to take some objects, she will take a copy of each available object. In addition, essential monotonicity applies also to situations where all the objects are "bads"; one replaces the copies of every object by the copies of "not taking this object." Section 1.1 illustrates the applicability of essentially monotonicity with specific examples of each case.

In order to show the existence of a desirable random assignment rule, we extend the Probabilistic Serial rule (Bogomolnaia and Moulin (2001)) and the Random Serial Dictatorship rule (Abdulkadiroğlu and Sönmez (1998)) to the setting with bundles. Proposition 2 and 3 study respectively the performance of these two rules on the universal domain. In particular, the random serial dictatorship rule for bundles, henceforth the RSDB rule, satisfies sd-strategy-proofness, decomposability, and equal treatment of equals but violates sd-efficiency. On the contrary, the probabilistic serial rule for the bundles, henceforth the PSB rule, satisfies sd-efficiency and equal treatment of equals but violates sd-strategy-proofness and decomposability.[2]

Theorem 2 proves that these two rules are equivalent on the essentially monotonic domain, which then implies the desired existence result, stated in Corollary 1. It is surprising that these two rules become equivalent on such a large domain in the setting with bundles. This is in sharp contrast to the disparity these two rules display in the setting with objects. Another feature of our result is that both rules degenerate to a constant rule. Here too, the fact that sd-efficiency is preserved by a simple constant assignment on such a large domain is rare in mechanism design and comes as a surprise.

In addition to the existence result on the essentially monotonic domain, following the literature, we demonstrate an impossibility result on the universal domain.[3] In particular, we show that, if there are more than three critical bundles, there is no sd-strategy-proof, sd-efficient, and sd-envy-free rule on the universal domain, where sd-envy-freeness is a fairness axiom stronger than equal treatment of equals and requires that every agent weakly prefers her own lottery to any other's.

The final part of our analysis deals with the issue of decomposability. We provide a necessary condition for decomposability, which is used to show that the random assignment generated by the PSB rule is in general not decomposable. Moreover, we transform the problem of decomposing a given random assignment to a series of maximum flow problems. Since maximum flow problems are amenable to numerical analysis, this reduction might be helpful for the decomposition of random assignments.

The remainder of this section contains two subsections. Section 1.1 presents a specific example of the formulation and the applicability of essentially monotonic domains. Section 1.2 provides a brief literature review. Following the introduction, Section 2 defines formally the model and axioms we study. Section 3 characterizes sd-efficient random assignments. Section 4

[2] Liu (2017) proved that the PS rule in the setting with objects is sd-strategy-proof on the sequentially dichotomous domains. We will show that when the PS rule and the sequentially dichotomous domains are extended naturally to the setting with bundles, this possibility fails.

[3] See Bogomolnaia and Moulin (2001), Kasajima (2013), Chang and Chun (2017), and Liu and Zeng (2017).

consists of five subsections. Subsection 4.1 and 4.2 define and examine, respectively, the RSDB rule and the PSB rule. Subsection 4.3 introduces the domain of essentially monotonic preferences. Subsection 4.4 shows the equivalence between the two rules on the essentially monotonic domain. Subsection 4.5 presents a general impossibility on the universal domain. Section 5 studies the decomposability of random assignments and Section 6 concludes.

1.1 A Leading Example

We present in this subsection an example that illustrates the scope of our study.

Example 1. A local government has developed a public housing project, which provides 50 units of standardized apartments, 50 parking slots, and 30 bicycle slots. These objects need to be fully allocated among 100 eligible applicants without any money transfers. ■

The objects in the example above are publicly financed and consequently free disposal is not an option as the objects ought not to be wasted. Hence a treatment that replicates the rules used in the object assignment literature (see for example Pápai (2000a) and Bogomolnaia and Moulin (2001)) are not appropriate since a central assumption there is that each agent receives at most one object, which implies for our problem that finally 30 objects need to be discarded.

Another way to allocate these objects is to do so via independent committees, one for each type of the objects. However, given that the agents' preferences may reflect complementarity and substitution, the efficiency losses incurred by this method may be significant.

It is therefore of interest to investigate the allocation of objects in bundles. Moreover, due to the indivisibility of the objects, agents reporting the same preference will potentially be treated differently. In order to achieve fairness, we resort to random assignments. In particular, every agent receives a lottery on bundles.

There are in total 8 distinct bundles. In order to simplify the notation, we denote an apartment as a, a parking slot as p, and a bicycle slot as b. In addition, a bundle is denoted simply as a sequence of these alphabets rather than its set format. For example, the whole bundle is denoted as apb rather than $\{a, p, b\}$.

Since objects are given to the agents for free, it might be reasonable to assume that a bundle containing more objects is better. This is captured by a classical preference restriction, monotonicity, which requires that a bundle contained in another bundle is less preferred to this later bundle. A typical monotonic preference is as follows.

$$P_i: \quad apb \succ ap \succ ab \succ pb \succ a \succ p \succ b \succ \emptyset$$

Our preference restriction, essential monotonicity, weakens monotonicity in the sense that it is required only for every critical bundle that if a bundle is contained in a critical bundle, it is less preferred. We now proceed to identify the critical bundles for this example. We assume that agents are lined up and required to take bundles one by one. Moreover, whenever an agent is called upon, she takes one copy of each available object. As for the current case, each of the first 30 agents will take the whole bundle apb, each of the next 20 agents will take ap, and each of the remaining agents will take the empty bundle $\emptyset$. Hence, there are in total three critical bundles. Essential monotonicity imposes three preference restrictions: (i) a bundle contained in apb is less preferred; (ii) a bundle contained in ap is less preferred; and (iii) a bundle contained in $\emptyset$ is less preferred.[4]

Hence the domain of essentially monotonic preferences includes not only all monotonic preferences, but also preferences like the one below, where the critical bundles are underlined.

$$P_i' : \quad \underline{apb} \succ \underline{ap} \succ ab \succ a \succ \underline{\emptyset} \succ b \succ p \succ pb$$

An agent having the above preference treats a bundle acceptable (better than getting nothing) if and only if it contains an apartment. Moreover, she treats the bicycle slots and parking slots as benefits if she gets an apartment. However, she treats these as bads otherwise.[5] The classical monotonicity requirement captures only complete complementarity: the more the better no matter the status quo. As shown by P_i, getting additional objects is always preferred. However, our notion of essential monotonicity captures also "partial complementarity." As shown by P_i', getting an apartment is always preferred. But whether getting a bicycle slot and a parking slot is preferred depends on whether the agent already gets an apartment.

As mentioned earlier, the set of critical bundles varies according to capacities. When the objects have the same capacity, there are only two critical bundles: the whole bundle and the empty bundle. In this case, the only restriction imposed on essentially monotonic preferences is that the whole bundle is the favorite. To the other extreme, when the objects have capacities different from each other, there are $m + 1$ critical bundles, where m is the number of object types. To summarize, the greater the variation in the capacities, the greater the number of critical bundles and hence the greater the number of restrictions on the essentially monotonic preferences.

Now consider the following alternative scenario, where we seek to allocate three types of tasks, a, b, and c respectively, among a given set of agents. In particular, assume that 50 working days of task a, 50 working days of task b, and 30 working days of task c have to be allocated among 100 team members. In this situation, objects appear to be "bads" and it would appear that the essential monotonicity requirement is not applicable to such situations as it would require, in the very least, that the grand bundle is the favorite. But the following preference, which seems to be reasonable in this situation, treats the empty set as the best.

$$P'' : \emptyset \succ a \succ b \succ c \succ ab \succ ac \succ bc \succ abc$$

However, the following observation makes essential monotonicity applicable. We need only to treat the following as the objects to be allocated: 50 copies of "not serving a working day of task a," 50 copies of "not serving a working day of task b," and 70 copies of "not serving a working day of task c." These imaginary objects are denoted as $\bar{a}$, $\bar{b}$, and $\bar{c}$ respectively. For

[4]Notice that the last restriction is vacuous. However, we still present it as a restriction in order to simplify definition.

[5]Put otherwise, receiving a parking slot or a bicycle slot without living nearby is costly. This is plausible for two reasons. First, objects obtained from a publicly financed project are usually not allowed to be used to make profit. So it would be very difficult to benefit from renting them out. Second, receiving a parking slot in a project may probably exclude the agent from getting a parking slot in some future projects, which might assign an apartment to her.

this new problem, the critical bundles are $\bar{a}\bar{b}\bar{c}$, $\bar{c}$, and $\emptyset$. Then the preference P'' above can be translated to $\bar{P}''$, which is essentially monotonic for the imaginary problem.

$$P'' : \underline{\bar{a}\bar{b}\bar{c}} \succ \bar{b}\bar{c} \succ \bar{a}\bar{c} \succ \bar{a}\bar{b} \succ \underline{\bar{c}} \succ \bar{b} \succ \bar{a} \succ \underline{\emptyset}$$

Hence, once these imaginary objects are allocated, the tasks are allocated automatically. Moreover, essential monotonicity should be a reasonable preference restriction for these imaginary objects.

Given a random assignment problem of bundles with essentially monotonic preferences, we can simply identify the critical bundles and then equally allocate these bundles to the agents. This method turns out to be equivalent to both the PS rule and the RSD rule, extended to the bundle setting. The point will become clear as we proceed.

1.2 Related Work

Hylland and Zeckhauser (1979) were the first to introduce randomization to the allocations of indivisible objects. Two decades later, Abdulkadiroğlu and Sönmez (1998) proved the random serial dictatorship rule as equivalent to the randomization over core allocations, which explains the popularity of this rule in realistic applications. Then Bogomolnaia and Moulin (2001) proposed to evaluate its efficiency ex ante rather than ex post and pointed out that the RSD rule is sd-inefficient. Instead, they introduced the simultaneous eating algorithm to generate sd-efficient random assignments. Bogomolnaia and Moulin (2001) showed also that the RSD rule is sd-strategy-proof but not sd-efficient while the PS rule is sd-efficient but not sd-strategy-proof. Pápai (2000b) and Pápai (2001) consider allocations in bundles rather than in individual objects. Respectively, Pápai (2001) imposes no preference restriction while Pápai (2000b) imposed monotonicity. However, these studies did not adopt randomization.

More closely related are the recent papers on random assignment of bundles, among which Budish et al. (2013), Nguyen et al. (2016), Akbarpour and Nikzad (2017) focused on decomposability of a random assignment. In particular, Budish et al. (2013) generalize the Birkhoff-von Neumann theorem to a larger class of probabilistic matrices. Akbarpour and Nikzad (2017) further stated that if a subset of feasibility restrictions can be viewed as "soft," the class of decomposable probabilistic matrices can be further enlarged. Relatively, Nguyen et al. (2016) studied the approximation of decomposition and stated that if agents can not take bundles containing more than k objects, every random assignment can be decomposed approximately in the sense that each object is over-allocated by at most $k-1$ units, ex post. We will discuss these papers in greater detail in Section 5.

Except for these three papers, there are others studying the designing of desirable random assignment rules, including Budish (2011), Sönmez and Ünver (2010), and Budish and Cantillon (2012). Our study differs from theirs mainly on the following three points. First, they all assume free disposal, which is a reasonable assumption given that they were studying course allocations in universities. Second, they focused on the so-called "pseudo-market" approach which endows the agents with pseudo money and then the mechanism mimics the market equilibrium. Our approach is different and the results are different since we identify a preference restriction to prove the existence of a desirable rule with no free disposal in the model.

2 Model and Axioms

Let $I \equiv \{1, \cdots, n\}$ denote a finite set of agents and let X denote a finite set of objects. Let in addition $m \equiv |X|$. Assume $n \geqslant 2$ and $m \geqslant 2$. In order to incorporate situations where some objects are physically identical, we allow an object to have multiple copies. For each $x \in X$, the capacity of x is a positive integer q_x, which denotes the number of its copies. We assume $q_x \in \{1, 2, \cdots, n-1\}$ so that the number of copies is smaller than the number of agents.[6] The capacities are collected in a vector $q = (q_x)_{x \in X}$.

A bundle of objects is a subset of X. The set of bundles is hence the power set 2^X and denoted as $\mathcal{X}$. Note that our definition of a bundle does not allow it to contain more than one copy of each object. Throughout the paper we denote objects with lowercase English alphabets and denote bundles with uppercase English alphabets, i.e., $a, b, c, x, y, z \in X$ and $A, B, C \in \mathcal{X}$. In addition, we usually denote the bundle $\{a, b, c\}$ simply as abc.

Each agent $i \in I$ is assumed to have a strict preference P_i on bundles, i.e., a linear order on $\mathcal{X}$. Following the convention, we denote $A \ R_i \ B$ if and only if either $A = B$ or $A \ P_i \ B$. The set of all strict preferences is denoted as $\mathbb{P}$ and referred to as the universal domain. Let $\mathbb{D} \subset \mathbb{P}$ be a nonempty subset of the universal domain. We treat this given subset as the set of admissible preferences and call it **the domain** of the problem. Given an arbitrary nonempty subset of bundles $\bar{\mathcal{X}} \subset \mathcal{X}$ and an arbitrary preference $P_i \in \mathbb{P}$, denote $r_k(P_i, \bar{\mathcal{X}})$ as the k-th ranked bundle in $\bar{\mathcal{X}}$ according to P_i, i.e., $|\{A \in \bar{\mathcal{X}} : A \ R_i \ r_k(P_i, \bar{\mathcal{X}})| = k$.

A deterministic assignment can be presented as a matrix, whose rows are associated with agents and columns associated with bundles. The elements are either zeros or ones, where "one" means the corresponding agent gets the corresponding bundle and "zero" means she does not. Each agent gets exactly one bundle, which means every row of the matrix has one non-zero element. Notice that this does not mean every agent will get some object, since the empty set is also treated as a bundle, i.e., $\emptyset \in \mathcal{X}$. In addition, an object $x \in X$ with capacity q_x is allocated to exactly q_x agents. We therefore impose **no free disposal**, which is a well justified assumption for relevant applications. Deterministic assignments are formally defined below.

Definition 1. *A **deterministic assignment** is a matrix $D \in \{0, 1\}^{I \times \mathcal{X}}$ such that*

1. $\forall \, i \in I$: $\sum_{A \in \mathcal{X}} D_{iA} = 1$,
2. $\forall \, x \in X$: $\sum_{i \in I, x \in A} D_{iA} = q_x$.

The set of deterministic assignments is denoted $\mathcal{D}$. The following is an example.

Example 2. *Let $I = \{1, 2, 3\}$, $X = \{a, b\}$, $q_a = 1$, and $q_b = 2$. Figure 1 below depicts a deterministic assignment which specifies that agent 1 gets bundle ab, agent 2 gets nothing, and agent 3 gets the remaining b.* ■

If one restricts attention to deterministic assignments, one would expect that, in general, the agents with the same preference will be treated unequally. For instance, in the setting of

[6]Since we allow the number of objects, i.e., m, be arbitrary, an object with more than $n-1$ copies can always be treated as several distinct objects, each of which has a capacity smaller than $n-1$.

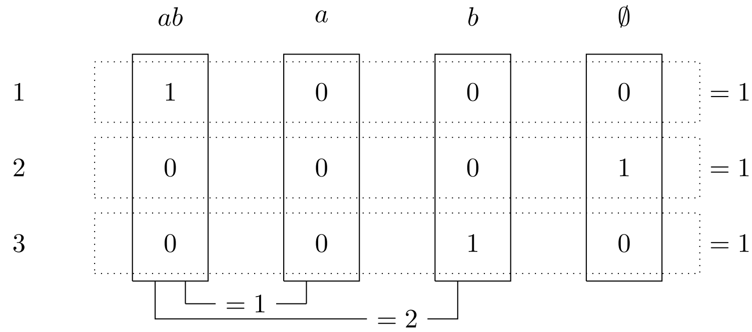


Figure 1: A Deterministic Assignment

Example 2, if all the agents perceive the bundle ab as the best and b the second best, it is impossible to treat them equally using only deterministic assignments.

To allow for greater flexibility in design to deal with the fairness issue, we allow the elements of an assignment to be fractional numbers between zero and one, as below.

Definition 2. *A* ***random assignment*** *is a matrix* $L \in [0,1]^{I \times \mathcal{X}}$ *such that*

1. $\forall\, i \in I$: $\sum_{A \in \mathcal{X}} L_{iA} = 1$,
2. $\forall\, x \in X$: $\sum_{i \in I, x \in A} L_{iA} = q_x$.

The set of random assignments is denoted $\mathcal{L}$. It is evident that $\mathcal{D} \subset \mathcal{L}$. The following is an example of a specific random assignment.

Example 3. *Let* $I = \{1,2,3\}$, $X = \{a,b\}$, $q_a = 1$, *and* $q_b = 2$. *Figure 2 below depicts a random assignment.* ■

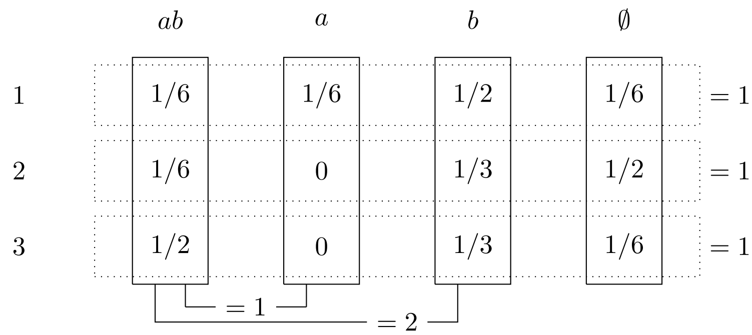


Figure 2: A Random Assignment

The fractional numbers in a random assignment are interpreted as the probability of the corresponding agent getting the corresponding bundle. Hence a row associated to agent i, denoted L_i, gives the lottery over bundles for agent i. As in the above random assignment, L_2 specifies that agent 2 will get bundle ab with probability $1/6$, b with probability $1/3$, and empty bundle with probability $1/2$.

For deterministic assignments, condition 2 in the definition simply imposes ex post feasibility. For random assignments, the situation is more complicated. To fully interpret it, we need to introduce another notion below.

Definition 3. *A random assignment $L \in \mathcal{L}$ is* ***decomposable*** *if there is a lottery over deterministic assignments $\beta \in \Delta(\mathcal{D})$ such that*

$$L = \sum_{D \in \mathcal{D}} \beta(D) \cdot D,$$

where $\beta(D)$ denotes the probability lottery β assigns to D.

Such a lottery β is called a **decomposition** of L. Generally a decomposable random assignment may have multiple decompositions. The following is an example of a decomposition.

Example 4. *The random assignment L in Example 3 can be decomposed as follows*

$$\begin{pmatrix} & ab & a & b & \emptyset \\ 1: & 1/6 & 1/6 & 1/2 & 1/6 \\ 2: & 1/6 & 0 & 1/3 & 1/2 \\ 3: & 1/2 & 0 & 1/3 & 1/6 \end{pmatrix} = 1/2 \begin{pmatrix} & ab & a & b & \emptyset \\ 1: & 0 & 0 & 1 & 0 \\ 2: & 0 & 0 & 0 & 1 \\ 3: & 1 & 0 & 0 & 0 \end{pmatrix} + 1/6 \begin{pmatrix} & ab & a & b & \emptyset \\ 1: & 1 & 0 & 0 & 0 \\ 2: & 0 & 0 & 1 & 0 \\ 3: & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$+ 1/6 \begin{pmatrix} & ab & a & b & \emptyset \\ 1: & 0 & 0 & 0 & 1 \\ 2: & 1 & 0 & 0 & 0 \\ 3: & 0 & 0 & 1 & 0 \end{pmatrix} + 1/6 \begin{pmatrix} & ab & a & b & \emptyset \\ 1: & 0 & 1 & 0 & 0 \\ 2: & 0 & 0 & 1 & 0 \\ 3: & 0 & 0 & 1 & 0 \end{pmatrix}$$

■

For a decomposable random assignment, Condition 2 in Definition 2 requires that, for x, the expected number of its copies that will be assigned to agents, through feasible deterministic assignments is exactly q_x. In this sense, condition 2 imposes ex ante feasibility. We will sometimes call a random assignment feasible in order to emphasize the conditions in Definition 2.

In the setting with objects, a random assignment is a bi-stochastic matrix, where ex ante feasibility is modeled as the requirement that every column sums to one. The Birkhoff-von Neumann theorem guarantees that every random assignment of objects is decomposable. However, in the setting with bundles, ex ante feasibility becomes complicated: it is not that every column sums to one independently but that columns sum to certain integers group by group in a combinatorial fashion. Such complexity in feasibility makes decomposability a difficult problem, which will be discussed in detail in Section 5.

When we restrict our attention to decomposable random assignments, expressing a random assignment as a convex combination of feasible deterministic assignments is equivalent to expressing it as the matrix defined by Definition 2. However, our definition facilitates comparisons of random assignments, since for an agent to compare two random assignments, it suffices to compare two rows directly. In addition, our definition incorporates non-decomposable random assignments, which can presumably be approximated by decomposable ones and are hence of some interest of their own right.

By allowing for some non-decomposable random assignments, one is allowed to design a random assignment rule which may choose non-decomposable random assignments but perform well in other dimensions. For example, the probabilistic serial rule, extended to the setting with bundles, selects some non-decomposable random assignments. However, this rule guarantees efficiency.

A random assignment rule is formally defined as a mapping which selects a random assignment for every profile of admissible preferences.

Definition 4. *A* ***random assignment rule*** *is a mapping* $\varphi : \mathbb{D}^n \to \mathcal{L}$.

The remainder of the section introduces four axioms that we impose on a desirable random assignment rule. The first axiom concerns itself with decomposability, which ensures that once a random assignment is selected, it can be decomposed as a lottery over deterministic assignments. The other three are normative and deal with respectively fairness, incentive compatibility, and efficiency.

We call a random assignment rule decomposable if it selects only among decomposable random assignments. Formally, a random assignment rule $\varphi : \mathbb{D}^n \to \mathcal{L}$ is **decomposable** if $\varphi(P)$ is decomposable for every $P \in \mathbb{D}^n$.

In addition to decomposability, we impose three normative axioms on a desirable random assignment rule. The first deals with fairness and requires that whenever two agents report the same preference, they get the same lottery. Formally, a rule $\varphi : \mathbb{D}^n \to \mathcal{L}$ satisfies **equal treatment of equals** (or **ETE**) if for all $P \in \mathbb{D}^n$, $[P_i = P_j] \Rightarrow [L_i = L_j]$.

The second deals with efficiency and the third deals with incentive compatibility. However, both of these require an assumption on how an agent compares lotteries when she is identified by a preference on bundles. We thus need to extend a preference P_i over bundles $\mathcal{X}$ to a preference over lotteries in $\Delta(\mathcal{X})$. Following the standard approach, we adopt the stochastic dominance extension, which assumes that a lottery $L_i \in \Delta(\mathcal{X})$ is at least as good as $L'_i \in \Delta(\mathcal{X})$ if, for each bundle $A \in \mathcal{X}$, the probability of getting a bundle that is at least as good as A given by L_i is no less than that given by L'_i.[7] Formally,

Definition 5. *Given* $P_i \in \mathbb{P}$*,* $L_i \in \Delta(\mathcal{X})$ ***stochastically dominates*** $L'_i \in \Delta(\mathcal{X})$*, denoted as* $L_i \ P_i^{sd} \ L'_i$*, if for all* $B \in \mathcal{X}$

$$\sum_{A R_i B} L_{iA} \geqslant \sum_{A R_i B} L'_{iA}.$$

With the stochastic dominance extension, we define the remaining two axioms. An assignment L is sd-efficient at $P \in \mathbb{D}^n$ if there exists no $L' \in \mathcal{L}$ that Pareto dominates L, i.e., $L' \neq L$ and $L'_i \ P_i^{sd} \ L_i$ for all $i \in I$. Accordingly, a rule $\varphi : \mathbb{D}^n \to \mathcal{L}$ is **sd-efficient** if $\varphi(P)$ is sd-efficient at P, for all $P \in \mathbb{D}^n$. We address the general question of whether a random assignment is sd-efficient at a given profile in Section 3. Finally, a rule is sd-strategy-proof if truth-telling is always a weakly dominant strategy in the associated preference revelation game. Formally, a rule $\varphi : \mathbb{D}^n \to \mathcal{L}$ is **sd-strategy-proof** if for all $i \in I$, $P \in \mathbb{D}^n$, and $P'_i \in \mathbb{D}$, $\varphi_i(P_i, P_{-i}) \ P_i^{sd} \ \varphi_i(P'_i, P_{-i})$. We say a rule is **desirable** if it satisfies decomposability, sd-strategy-proofness, sd-efficiency, and equal treatment of equals.

[7] This assumption is equivalent to assuming that a lottery L_i is at least as good as L'_i if and only if, for every Bernoulli utility representing P_i, L_i gives an expected utility that is at least as high as that is given by L'_i.

3 Efficiency

We address in this section the question of whether a given random assignment is sd-efficient at a given preference profile. For the problem of random assignments of objects, Bogomolnaia and Moulin (2001) provided two characterizations of sd-efficient assignments. The first says that a random assignment is sd-efficient at a profile if and only if a particular relation on objects is acyclic. The second is more mechanical and shows that at a profile, all sd-efficient random assignments can be found by the simultaneous eating algorithm with varying eating speeds. For the assignment problem of bundles, we find that neither characterization is true. In particular, acyclicity, while still necessary, is not sufficient. A new condition called unbalancedness is provided and proved equivalent to sd-efficiency in the current setting. We provide two more conditions, which are natural modifications of unbalancedness and are referred to as respectively strong unbalancedness and weak unbalancedness. Next, we extend the simultaneous eating algorithm to the bundle setting and prove that all random assignments generated by this algorithm (with varying eating speeds) are sd-efficient. However, surprisingly we find that there exists an sd-efficient random assignment at some profile that can not be generated by the simultaneous eating algorithm with any eating speed.[8]

We begin with a modified definition of acyclicity (Bogomolnaia and Moulin (2001)).

Definition 6. *A random assignment $L \in \mathcal{L}$ is* ***acyclic*** *at $P \in \mathbb{P}^n$ if and only if the relation $\tau(P, L)$ on $\mathcal{X}$ is acyclic where $A\ \tau(P, L)\ B \Leftrightarrow \exists i \in I$ such that $B\ P_i\ A$ and $L_{iA} > 0$.*

The next example shows that acyclicity is no longer sufficient to guarantee sd-efficiency.

Example 5. *Let $A = \{a, b, c\}$, $q = (1, 1, 1)$, $I = \{1, 2\}$. Let the preferences of two agents be*

$$\begin{array}{lllllllll} P_1: & c & a & ab & b & \emptyset & bc & ac & abc \\ P_2: & a & c & ab & b & \emptyset & bc & ac & abc \end{array}$$

Consider random assignments L and L' below.

	c	a	ab	b	$\emptyset$	bc	ac	abc
L_1:	0	0	0.2	0	0.3	0	0	0.5
L_2:	0.2	0	0	0	0.5	0	0	0.3

	c	a	ab	b	$\emptyset$	bc	ac	abc
L'_1:	0.2	0	0	0.2	0.1	0	0	0.5
L'_2:	0	0.2	0	0	0.5	0	0	0.3

We claim that L above is acyclic at P. To see this, notice that if the relation $\tau(P, L)$ has a cycle, it must involve a preference reversal across two agents' preferences. According to P, agents' preferences coincide except between a and c. Hence a cycle of $\tau(P, L)$ requires at the same time $a\ \tau(P, L)\ c$ and $c\ \tau(P, L)\ a$. We show however $a\ \tau(P, L)\ c$ is not true. To see this, notice that agent 1 who prefers c to a has no positive probability of a and that agent 2 does not prefer c to a. Hence the relation $\tau(P, L)$ is acyclic. However L is not sd-efficient at P since it is Pareto dominated by L': $L' \neq L$, $L'_1\ P_1^{sd}\ L_1$, and $L'_2\ P_2^{sd}\ L_2$. ■

[8]Due to its complexity, the related context is in Appendix A.2. The readers are suggested to read the Subsection 4.2, where the probabilistic serial rule for bundles is defined, before the definition of the simultaneous eating algorithm with varying eating speeds.

The failure of the acyclicity to suffice for sd-efficiency occurs as we are able to implement a sequence of probability transfers starting at L and leading to a feasible Pareto improvement. In particular, by comparing L and L' in the above example, we identify three probability transfers which construct L' from L, illustrated by Figure 3.

$$
\begin{array}{lcccccccc}
 & c & a & ab & b & \emptyset & bc & ac & abc \\
L_1\colon & 0 & 0 & 0.2 & 0 & 0.3 & 0 & 0 & 0.5 \\
 & \multicolumn{3}{c}{\scriptstyle \alpha(1,\,ab,\,c)\,=\,0.2} & \multicolumn{3}{c}{\scriptstyle \alpha(1,\,ab,\,c)\,=\,0.2} & & \\
L_2\colon & 0.2 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0.3 \\
 & \multicolumn{2}{c}{\scriptstyle \alpha(2,\,c,\,a)\,=\,0.2} & & & & & &
\end{array}
$$

Figure 3: Probability Transfer System α

In particular, let $\alpha(1, ab, c) = 0.2$ denote the probability transfer of 0.2 from $(1, ab)$ to $(1, c)$ and let $\bar{L}$ denote the matrix resulting from such a probability transfer on L. Since c is preferred to ab by agent 1, she would prefer such a transfer. However, $\bar{L}$ is not a feasible random assignment. Specifically, $\sum_{i\in I,a\in A} \bar{L}_{iA} = q_a - 0.2$, $\sum_{i\in I,b\in A} \bar{L}_{iA} = q_b - 0.2$, and $\sum_{i\in I,c\in A} \bar{L}_{iA} = q_c + 0.2$. The net influence on the feasibility is as shown by the second column in the following table.

	$\alpha(1, ab, c) = 0.2$	$\alpha(1, \emptyset, b) = 0.2$	$\alpha(2, c, a) = 0.2$	Total
a	-0.2	0	$+0.2$	0
b	-0.2	$+0.2$	0	0
c	$+0.2$	0	-0.2	0

Next, we denote the remaining two transfers as respectively $\alpha(1, \emptyset, b) = 0.2$ and $\alpha(2, c, a) = 0.2$ and implement them successively starting from $\bar{L}$ to obtain L'. The third and fourth columns in the above table summarize the influence of these two transfers on feasibility. We see that the influence of the aforementioned transfers cancel out on each row, making L' a feasible random assignment.

To formalize the observation above, let $\mathcal{T} = I \times \mathcal{X} \times \mathcal{X}$. Then a **system of probability transfers** can be represented by a mapping $\alpha : \mathcal{T} \to \mathbb{R}_+$ that specifies for each triple (i, A, B) a non-negative number $\alpha(i, A, B)$, which denotes a probability transfer from (i, A) to (i, B). Given a random assignment $L \in \mathcal{L}$, a transfer system will construct a new matrix, denoted L', of size $|I| \times |\mathcal{X}|$. Formally $\forall j \in I$ and $C \in \mathcal{X}$,

$$L'_{jC} = L_{jC} + \sum_{\{(i,A,B)\in\mathcal{T}:i=j,B=C\}} \alpha(i, A, B) - \sum_{\{(i,A,B)\in\mathcal{T}:i=j,A=C\}} \alpha(i, A, B).$$

Generally such a matrix L' is not a feasible random assignment. In the definition below, we focus on a particular class of systems which not only construct feasible random assignments but also ensure that the assignments constructed dominate the original L at P.

Definition 7. *An assignment $L \in \mathcal{L}$ is* ***unbalanced*** *at $P \in \mathbb{P}^n$ if there is no $\alpha : \mathcal{T} \to \mathbb{R}_+$ s.t.*

(i) $\alpha(i, A, B) > 0$ *implies* $L_{iA} > 0$ *and* $B \; P_i \; A$,

(ii) $\forall x \in X : \sum_{\{(i,A,B)\in\mathcal{T}:x\in B\}} \alpha(i, A, B) = \sum_{\{(i,A,B)\in\mathcal{T}:x\in A\}} \alpha(i, A, B)$.

We say L is ***balanced*** *at P if it is not unbalanced at P.*

The following theorem states that unbalancedness characterizes sd-efficiency.

Theorem 1. *Given $P \in \mathbb{P}^n$ and $L \in \mathcal{L}$, L is sd-efficient at P iff L is unbalanced at P.*

The formal proof is in Appendix A.1. Here we work through the logic of the proof using an example.

The necessity part was illustrated by Example 5 where we showed that L was balanced at P by demonstrating the existence of an $\alpha : \mathcal{T} \to \mathbb{R}_+$ which satisfies (i) and (ii). More generally, if a configuration like the one in Example 5 obtains, we can guarantee the existence of an $\alpha : \mathcal{T} \to \mathbb{R}_+$ and by implementing the probability transfers multiplied by a small positive number, we can guarantee a feasible random assignment L' which Pareto dominates L.

We now turn to the sufficiency argument. Suppose another random assignment L' dominates L at P. We need to find a mapping $\alpha : \mathcal{T} \to \mathbb{R}_+$ that satisfies (i) and (ii). To do this, we first pick an arbitrary mapping $\beta : \mathcal{T} \to \mathbb{R}_+$ that constructs L' from L.[9] Notice that by the fact that L' is a feasible random assignment, (ii) is satisfied automatically. If β satisfies in addition (i), it serves as the desired α directly. However that is in general not the case since it is possible that $\beta(i, A, B) > 0$, $L_{iA} = 0$ and(or) that $\beta(i, A, B) > 0$, $A \; P_i \; B$.

Given an arbitrary such β, we construct a desired α from it in two steps. Each step consists of finitely many updates upon β. In particular, the first step deals with the instances where $\beta(i, A, B) > 0$ with $L_{iA} = 0$ and finds an intermediary mapping $\gamma : \mathcal{T} \to \mathbb{R}_+$ that constructs L' from L and satisfies the condition $\gamma(i, A, B) > 0$ implies $L_{iA} > 0$. The second step deals with the instances where $\gamma(i, A, B) > 0$ with $A \; P_i \; B$ and finds the desired α. The following example illustrates the procedure.

Example 6. *Let P, L, and L' be as in Example 5, where we already stated that L' dominates L at P. Note that if a transfer system constructs L' from L, it satisfies (ii). Hence it suffices to construct a transfer system $\alpha : \mathcal{T} \to \mathbb{R}_+$ which constructs L' from L and satisfies (i). Notice that L' can be constructed from L by $\beta : \mathcal{T} \to \mathbb{R}_+$ illustrated by Figure 4.*

In the first step, notice that $\beta(1, a, c) = 0.2$ while $L_{1a} = 0$. Therefore β can not serve as a desired probability transfer system. To deal with this undesired violation of (i) of Definition 7, we identify $\beta(1, \emptyset, a) = 0.2$, which transfers probability to $(1, a)$. Then we update β to γ illustrated by Figure 5. It is evident that L' is constructed from L by γ.[10]

[9] Since both L and L' are feasible random assignments, such a β exists. In particular, for each $i \in I$, if there is a bundle A such that $L'_{iA} > L_{iA}$, then there is a subset of bundles $\{B_k : k = 1, \cdots, K\} \subset \mathcal{X} \backslash \{A\}$ such that $L'_{iB_k} < L_{iB_k}$ and $\sum_{k=1}^{K} L_{iB_k} - L'_{iB_k} \geqslant L'_{iA} - L_{iA}$.

[10] By the fact that L' is constructed from L by β, such an update always exists. In general, to get such a γ, we need multiple updates. For details, please refer to the formal proof.

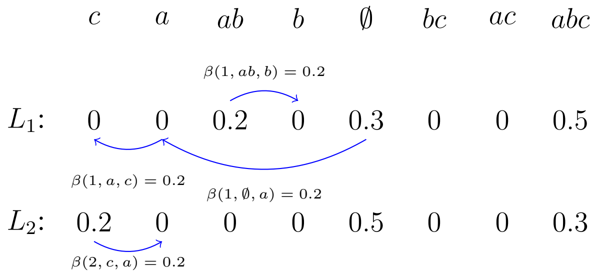


Figure 4: Probability Transfer System β

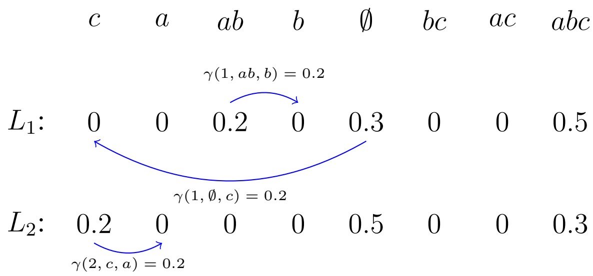


Figure 5: Probability Transfer System γ

In the second step, notice that $\gamma(1, ab, b) = 0.2$ while $ab\ P_1\ b$. To deal with this violation of (i) of Definition 7, we identify $\gamma(1, \emptyset, c) = 0.2$ and $c\ P_1\ ab\ P_1\ b\ P_1\ \emptyset$. Then we update γ to α by replacing these two probability transfers by $\alpha(1, \emptyset, b) = 0.2$ and $\alpha(1, ab, c) = 0.2$, as illustrated by Figure 6.[11]

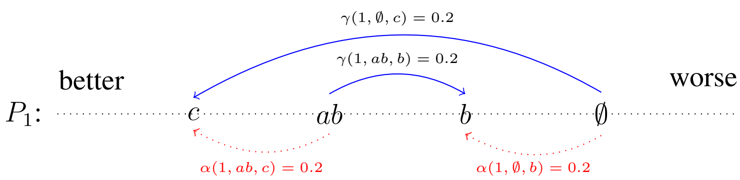


Figure 6: Construction of α from γ

It is then easy to check that L' is constructed from L by α and that α satisfies (i). ■

We proceed by providing two more conditions, respectively strong unbalancedness and weak unbalancedness. Weak unbalancedness is stronger than acyclicity. Strong unbalancedness will be used to show that the random assignment generated by the simultaneous eating algorithm with any eating speed is sd-efficient in Appendix A.2.

In order to define these two conditions, we introduce some notation. Fix $P \in \mathbb{P}^n$, $L \in \mathcal{L}$, and a subset of triples $\mathcal{S} \subset \mathcal{T}$. For each object $x \in X$, let $d(x, \mathcal{S})$ count the triples $(i, A, B) \in \mathcal{S}$

[11] By the fact that $L'_1\ P_1^{sd}\ L_1$ and that L' is constructed from L by γ, we can always find such a sequence of bundles. However, generally, this sequence may involve more than four bundles. In addition, to get such a α, we usually need multiple updates. For details, please refer to the formal proof.

such that $x \in B$ and $x \notin A$. Recall that we interpret a triple (i, A, B) as a potential transfer from (i, A) to (i, B). Then for every such x, there is a positive influence on the feasibility of x when a positive transfer is implemented. (See Example 5.) In this way, $d(x, \mathcal{S})$ counts the instances of positive influence on the feasibility of x. Similarly, $s(x, \mathcal{S})$ counts the instances of negative influence on the feasibility of x. Formally

$$d(x, \mathcal{S}) \equiv \left|\{(i, A, B) \in \mathcal{S} : x \in B \backslash A\}\right| \text{ and } s(x, \mathcal{S}) \equiv \left|\{(i, A, B) \in \mathcal{S} : x \in A \backslash B\}\right|.$$

Definition 8. *An assignment $L \in \mathcal{L}$ is **strongly unbalanced** at $P \in \mathbb{P}^n$ if there is no $\mathcal{S} \subset \mathcal{T}$ s.t.*

(i) $(i, A, B) \in \mathcal{S}$ *implies* $L_{iA} > 0$ *and* $B\ P_i\ A$,

(ii) $\forall x \in X$: $d(x, \mathcal{S}) > 0 \Leftrightarrow s(x, \mathcal{S}) > 0$.

*We say L is **strongly balanced** at P if it is not strongly unbalanced at P.*

Definition 9. *An assignment $L \in \mathcal{L}$ is **weakly unbalanced** at $P \in \mathbb{P}^n$ if there is no $\mathcal{S} \subset \mathcal{T}$ s.t.*

(i) $(i, A, B) \in \mathcal{S}$ *implies* $L_{iA} > 0$ *and* $B\ P_i\ A$,

(ii) $\forall x \in X$: $d(x, \mathcal{S}) = s(x, \mathcal{S})$.

*We say L is **weakly balanced** at P if it is not weakly unbalanced at P.*

The following proposition establishes the logical relations among the definitions we have mentioned so far.

Proposition 1. *Strong unbalancedness* $\underset{\not\Leftarrow}{\Rightarrow}$ *sd-efficiency* $\underset{\not\Leftarrow}{\Rightarrow}$ *weak unbalancedness* $\underset{\not\Leftarrow}{\Rightarrow}$ *acyclicity.*

Proof. We prove the proposition in three steps, each of which breaks into two parts.

Step 1.1: Strong unbalancedness $\Longrightarrow$ sd-efficiency. Let $L \in \mathcal{L}$ be strongly unbalanced at $P \in \mathbb{P}^n$. Suppose L is not sd-efficient at P. Then by Theorem 1, L is balanced at P, which implies the existence of an $\alpha : \mathcal{T} \to \mathbb{R}_+$ such that (i) $\alpha(i, A, B) > 0$ implies $L_{iA} > 0$ and $B\ P_i\ A$, (ii) $\forall x \in X :\ \ \sum_{\{(i,A,B) \in \mathcal{T} : x \in B\}} \alpha(i, A, B) = \sum_{\{(i,A,B) \in \mathcal{T} : x \in A\}} \alpha(i, A, B)$.

Let $\mathcal{S} = \{(i, A, B) \in \mathcal{T} : \alpha(i, A, B) > 0\}$. Then by definition, $(i, A, B) \in \mathcal{S}$ implies $L_{iA} > 0$ and $B\ P_i\ A$. We claim that $\forall x \in X$: $d(x, \mathcal{S}) > 0 \Leftrightarrow s(x, \mathcal{S}) > 0$. Suppose not, let $x \in X$ be such that $d(x, \mathcal{S}) > 0$ and $s(x, \mathcal{S}) = 0$, with which a contradiction is identified below. (A contradiction can be identified analogously for the other case where $d(x, \mathcal{S}) = 0$ and $s(x, \mathcal{S}) > 0$.)

$$\begin{aligned}
&\sum_{\{(i,A,B) \in \mathcal{T} : x \in B\}} \alpha(i, A, B) - \sum_{\{(i,A,B) \in \mathcal{T} : x \in A\}} \alpha(i, A, B) \\
= &\sum_{\{(i,A,B) \in \mathcal{S} : x \in B\}} \alpha(i, A, B) - \sum_{\{(i,A,B) \in \mathcal{S} : x \in A\}} \alpha(i, A, B) \\
= &\sum_{\{(i,A,B) \in \mathcal{S} : x \in B \backslash A\}} \alpha(i, A, B) - \sum_{\{(i,A,B) \in \mathcal{S} : x \in A \backslash B\}} \alpha(i, A, B) \\
= &\sum_{\{(i,A,B) \in \mathcal{S} : x \in B\}} \alpha(i, A, B) - 0 > 0 : \text{ contradiction.}
\end{aligned}$$

In the above, the first two equations follow from definitions. The third equation follows from $s(x, \mathcal{S}) = 0$ and the last inequality follows from $d(x, \mathcal{S}) > 0$.

Step 1.2: Strong unbalancedness $\not\Leftarrow$ sd-efficiency. This is shown by the following example. Let $X = \{a, b, c\}$ and $q_x = 1$ for all $x \in X$. Let the preference profile P and the assignment L be as below.

$$\begin{array}{llll} P_1: & ab & c & \cdots \\ P_2: & c & b & \cdots \\ P_3: & c & a & \cdots \end{array} \qquad \begin{array}{lcccc} & a & b & c & ab \\ L_1: & 0 & 0 & 1 & 0 \\ L_2: & 0 & 1 & 0 & 0 \\ L_3: & 1 & 0 & 0 & 0 \end{array}$$

We show that L is sd-efficient at P. Suppose not, let L' dominate L.

To do this, we show $L'_1 = L_1$. Suppose not, $L'_1 \; P_1^{sd} \; L_1$ implies that $\exists \, \epsilon_1 \in (0, 1]$ s.t.

$$\begin{array}{lcccc} & a & b & c & ab \\ L'_1: & 0 & 0 & 1 - \epsilon_1 & \epsilon_1 \end{array}$$

Given this, $L'_2 \; P_2^{sd} \; L_2$ and $L'_3 \; P_3^{sd} \; L_3$ imply the existence of $\epsilon_2, \epsilon_3 \in [0, 1]$ such that

$$\begin{array}{lcccc} & a & b & c & ab \\ L'_2: & 0 & 1 - \epsilon_2 & \epsilon_2 & 0 \\ L'_2: & 1 - \epsilon_3 & 0 & \epsilon_3 & 0 \end{array}$$

Then feasibility requires $\epsilon_1 + (1 - \epsilon_3) = q_a = 1$ and $\epsilon_1 + (1 - \epsilon_2) = q_b = 1$, which imply $\epsilon_1 = \epsilon_2 = \epsilon_3$. This however implies $1 - \epsilon_1 + \epsilon_2 + \epsilon_3 \neq 1 = q_c$: contradiction.

Given $L'_1 = L_1$, feasibility implies $L'_{2c} = L'_{3c} = 0$ and hence $L'_2 \; P_2^{sd} \; L_2$ and $L'_3 \; P_3^{sd} \; L_3$ imply $L'_2 = L_2$ and $L'_3 = L_3$, which means $L = L'$: contradiction.

Next we show that L is strongly unbalanced at P. To do this, let $\mathcal{S} = \{(1, c, ab), (2, a, c), (3, b, c)\}$. Then it is easy to verify that L is strongly unbalanced at P: (i) $L_{1c} > 0$, $ab \; P_1 \; c$; $L_{2a} > 0$, $c \; P_2 \; a$; $L_{3b} > 0$, $c \; P_3 \; b$; (ii) $d(x, \mathcal{S}) > 0$ and $s(x, \mathcal{S}) > 0$ for all $x \in X$.

Step 2.1: Sd-efficiency $\Longrightarrow$ weak unbalancedness. By Proposition 1, it suffices to show unbalancedness $\Longrightarrow$ weak unbalancedness. Suppose an assignment $L \in \mathcal{L}$ be weakly balanced at $P \in \mathbb{P}^n$. Then there is a subset $\mathcal{S} \subset \mathcal{T}$ such that (i) $(i, A, B) \in \mathcal{S}$ implies $L_{iA} > 0$, $B \; P_i \; A$, and (ii) $\forall x \in X$: $d(x, \mathcal{S}) = s(x, \mathcal{S})$. In the following, we construct a mapping $\alpha : \mathcal{T} \to \mathbb{R}_+$ such that (i) $\alpha(i, A, B) > 0$ implies $L_{iA} > 0$, $B \; P_i \; A$, and (ii) $\forall x \in X$: $\sum_{\{(i,A,B) \in \mathcal{T} : x \in B\}} \alpha(i, A, B) = \sum_{\{(i,A,B) \in \mathcal{T} : x \in A\}} \alpha(i, A, B)$.

In particular, let $\alpha(i, A, B) = \epsilon$ for all $(i, A, B) \in \mathcal{S}$ and $\alpha(i, A, B) = 0$ otherwise, where ϵ is a small positive number. Then (i) is satisfied by definition and (ii) follows from the equation below.

$$
\begin{aligned}
\forall x \in X: & \sum_{\{(i,A,B)\in\mathcal{T}: x\in B\}} \alpha(i,A,B) - \sum_{\{(i,A,B)\in\mathcal{T}: x\in A\}} \alpha(i,A,B) \\
&= \sum_{\{(i,A,B)\in\mathcal{S}: x\in B\}} \alpha(i,A,B) - \sum_{\{(i,A,B)\in\mathcal{S}: x\in A\}} \alpha(i,A,B) \\
&= \sum_{\{(i,A,B)\in\mathcal{S}: x\in B\backslash A\}} \alpha(i,A,B) - \sum_{\{(i,A,B)\in\mathcal{S}: x\in A\backslash B\}} \alpha(i,A,B) \\
&= \epsilon \cdot d(x,\mathcal{S}) - \epsilon \cdot s(x,\mathcal{S}) = 0.
\end{aligned}
$$

Step 2.2: Sd-efficiency $\not\Longleftarrow$ weak unbalancedness. This is shown by the following example. Let $X = \{a, b, c, d\}$ and $q_x = 1$ for each $x \in X$. Let the preference profile P and the assignment L, L' be as below.

					a	b	c	d	$\emptyset$	ab		a	b	c	d	$\emptyset$	ab
$P_1:$	c	$\emptyset$	$\cdots$	$L_1:$	0	0	0	0	1	0	$L'_1:$	0	0	ϵ	0	$1-\epsilon$	0
$P_2:$	b	a	$\cdots$	$L_2:$	1	0	0	0	0	0	$L'_2:$	$1-\epsilon$	ϵ	0	0	0	0
$P_3:$	d	b	$\cdots$	$L_3:$	0	1	0	0	0	0	$L'_3:$	0	$1-2\epsilon$	0	2ϵ	0	0
$P_4:$	ab	c	$\cdots$	$L_4:$	0	0	1	0	0	0	$L'_4:$	0	0	$1-\epsilon$	0	0	ϵ
$P_5:$	$\emptyset$	d	$\cdots$	$L_5:$	0	0	0	1	0	0	$L'_5:$	0	0	0	$1-2\epsilon$	2ϵ	0

We verify that L is weakly unbalanced at P. Suppose not, let $\mathcal{S} \subset \mathcal{T}$ be a subset such that (i) $(i, A, B) \in \mathcal{S}$ implies $L_{iA} > 0$ and $B\ P_i\ A$, and (ii) $\forall\ x \in X$, $d(x, \mathcal{S}) = s(x, \mathcal{S})$. Notice that for each $i \in I$, L_i assigns the whole probability to the second ranked bundle according to P_i. Hence $\mathcal{S} \subset \{(1, \emptyset, c), (2, a, b), (3, b, d)), (4, c, ab), (5, d, \emptyset\}$. Next, by definition of weak unbalancedness, whichever triple $\mathcal{S}$ includes, it includes all five triples in order to make $d(x, \mathcal{S}) = s(x, \mathcal{S})$ for all $x \in X$. So $\mathcal{S} = \{(1, \emptyset, c), (2, a, b), (3, b, d), (4, c, ab), (5, d, \emptyset)\}$. However, $d(b, \mathcal{S}) = 2 \neq 1 = s(b, \mathcal{S})$: contradiction.

We verify second that L is not sd-efficient at P by showing that L is dominated by L'. Let $\epsilon \in (0, 1]$. Feasibility of L' is evident. Noticing that the change from L_i to L'_i for each $i \in I$ is moving some probability from the second ranked bundle to the top ranked, domination is easy to see. Hence L is not sd-efficient at P.

Step 3.1: Weak unbalancedness $\Longrightarrow$ acyclicity. Let $L \in \mathcal{L}$ be weakly unbalanced at $P \in \mathbb{P}^n$ but $\tau(P, L)$ has a cycle. Then let a cycle be as follows

$$A_1\ \tau(P,L)\ A_2\ \tau(P,L)\ A_3 \cdots A_{K-1}\ \tau(P,L)\ A_K\ \tau(P,L)\ A_1.$$

In addition, let i_k be such that $A_{k+1}\ P_{i_k}\ A_k$ and $L_{i_k A_k} > 0$. Let $\mathcal{S} = \{(i_k, A_k, A_{k+1}) : k = 1, \cdots, K\}$ with $A_{K+1} = A_1$. Fixing an arbitrary $x \in X$, we prove $d(x, \mathcal{S}) = s(x, \mathcal{S})$. If $x \notin A_k$ for all $k = 1, \cdots, K$. By definition $d(x, \mathcal{S}) = s(x, \mathcal{S}) = 0$. Otherwise, let $(i_{k-1}, A_{k-1}, A_k) \in \mathcal{S}$ be arbitrary such that $x \in A_k \backslash A_{k-1}$, it suffices to show the existence of $(i_{k+l}, A_{k+l}, A_{k+l+1}) \in \mathcal{S}$ such that $x \in A_{k+l} \backslash A_{k+l+1}$. By the fact that $A_1, \cdots, A_K$ forms a cycle, such a triple exists.

Step 3.2: Weak unbalancedness $\not\Leftarrow$ acyclicity. This is shown by the following example. Let $A = \{a, b, c\}$, $q = (1, 1, 1)$, $I = \{1, 2\}$. Let the preferences of two agents be

$$\begin{array}{llllllllll} P_1: & c & a & ab & b & \emptyset & bc & ac & abc \\ P_2: & a & c & ab & b & \emptyset & bc & ac & abc \end{array}$$

Consider a random assignment L below.

	c	a	ab	b	$\emptyset$	bc	ac	abc
L_1 :	0	0	0.2	0	0.3	0	0	0.5
L_2 :	0.2	0	0	0	0.5	0	0	0.3

We have stated in Example 5 that L is acyclic at P. However, let $\mathcal{S} = \{(1, ab, c), (1, \emptyset, b), (2, c, a)\}$. We have $c\ P_1\ ab$, $L_{1ab} > 0$, $b\ P_1\ \emptyset$, $L_{1\emptyset} > 0$, $a\ P_2\ c$, and $L_{2c} > 0$. In addition, by simply counting, $d(x, \mathcal{S}) = s(x, \mathcal{S}) = 1$ for all $x \in \{a, b, c\}$. Hence L is not weakly balanced at P. ■

4 An Existence Result

In the classical random assignment problem, two extensively studied rules are the random serial dictatorship rule (or RSD, see Abdulkadiroğlu and Sönmez (1998)) and the probabilistic serial rule (or PS, see Bogomolnaia and Moulin (2001)). It is well known that, in the classical random assignment model, both rules treat equals equally. The PS rule is sd-efficient but not sd-strategy-proof while the RSD rule is sd-strategy-proof but not sd-efficient. Due to their distinct properties, these two rules are treated as competing alternatives for applications.

This section consists of five subsections. Subsection 4.1 and 4.2 extend the RSD rule and the PS rule to the setting with bundles. Their performance in the current setting is summarized by Propositions 2 and 3. Subsection 4.3 introduces the domain of essentially monotonic preferences. Subsection 4.4 proves that the above mentioned rules are equivalent on this domain (Theorem 2). Such a surprising equivalence gives a desirable existence result: there exists a decomposable, sd-efficient, sd-strategy-proof, and equal-treatment-of-equals random assignment rule on the essentially monotonic domain. Lastly, Subsection 4.5 presents a general impossibility result on the universal domain (Proposition 4).

4.1 The Random Serial Dictatorship Rule for Bundles

In the classical random assignment model, the random serial dictatorship rule (Abdulkadiroğlu and Sönmez (1998)) is defined as the equally weighted average of serial dictatorship rules (Svensson (1999)), each of which is a deterministic rule parameterized by an ordering of agents. Such an ordering is defined as a one-to-one mapping $\sigma : \{1, \cdots, n\} \to I$, where $\sigma(1)$ denotes the agent ordered the first, $\sigma(2)$ the second, and so on. The corresponding serial dictatorship rule lets the agents pick their respectively favorite objects sequentially. In particular, $\sigma(1)$ gets her favorite object, $\sigma(2)$ gets her favorite within the remaining objects, and so on.

For the setting with bundles, a seemingly "natural" extension of the serial dictatorship rule is one where every agent takes a bundle rather than an object. However, the following example indicates that such an extension may specify an infeasible assignment because of its conflict with the no free disposal requirement.

Example 7. *Consider the situation where $X = \{a, b\}$, $q_a = q_b = 1$, and $I = \{1, 2\}$. Consider a preference profile where agents have the same preference: $a \succ \emptyset \succ ab \succ b$. Let σ be an ordering of agents such that $\sigma(1) = 1$ and $\sigma(2) = 2$.*

In the first step, the set of available bundles is $\{ab, a, b, \emptyset\}$. Hence agent 1 will take a. Then the set of available bundles for agent 2 will be $\{b, \emptyset\}$, from which agent 2 will choose $\emptyset$. Then the deterministic assignment will be $D_{1a} = D_{2\emptyset} = 1$. However, this is not feasible since b is not assigned. ■

To deal with the problem, we introduce for each object $x \in X$, an "opposite object", denoted as $\bar{x}$, and refer to it as "not x." In addition, we will say $\bar{x} \in A$ if $x \notin A$. For each $\bar{x}$, we define its capacity as $n - q_x$ and whenever an agent takes a bundle A which does not contain x, we deduct the available units of $\bar{x}$ by one. Accordingly, we define a bundle A as **available**, if for every $x \in X$, $x \in A$ implies that there are still some units of x available and $x \notin A$ implies that there is still some units of $\bar{x}$ available. This rules out the infeasible assignments seen in Example 7 since whenever an agent takes a bundle not containing x, the available units of each opposite object $\bar{x}$ will be less. So there is always a step when no more $\bar{x}$ will be available and hence some agent has to take x.

We present the serial dictatorship rule for bundles on an arbitrary domain below, where q_x^{v-1} and $q_{\bar{x}}^{v-1}$ denote respectively the available units of x and $\bar{x}$ for the v-th agent, which then defines as $\mathcal{X}^{v-1}$ the available bundles.

Definition 10. ***Serial dictatorship for bundles (SDB)*** *is a deterministic assignment rule $SDB^\sigma : \mathbb{D}^n \to \mathcal{D}$ parameterized by an ordering of agents $\sigma : \{1, 2, \cdots, n\} \to I$, such that given a preference profile $P \in \mathbb{D}^n$, $SDB^\sigma(P) = D$, specified by the following.*

Let $\mathcal{X}^0 = \mathcal{X}$, $q_x^0 = q_x$, and $q_{\bar{x}}^0 = n - q_x$.

For $v = 1, \cdots, n$,

$$D_{\sigma(v)A} = \begin{cases} 1 & \text{if } A = r_1(P_{\sigma(v)}, \mathcal{X}^{v-1}) \\ 0 & \text{otherwise} \end{cases};$$

$$\begin{aligned} q_x^v &= q_x^{v-1} - 1, \forall x \in r_1(P_{\sigma(v)}, \mathcal{X}^{v-1}); \\ q_{\bar{x}}^v &= q_{\bar{x}}^{v-1} - 1, \forall x \notin r_1(P_{\sigma(v)}, \mathcal{X}^{v-1}); \\ \mathcal{X}^v &= \mathcal{X}^{v-1} \backslash \{A \in \mathcal{X}^{v-1} : \exists x \in X \text{ s.t. } [x \in A, q_x^v = 0] \text{ or } [x \notin A, q_{\bar{x}}^v = 0]\}. \end{aligned}$$

To illustrate that the SDB rule is well-defined, we present the following example.

Example 8. *Consider the setting of Example 7. The capacities of the objects are as follows:*

$$q_a^0 = 1, \quad q_b^0 = 1, \quad q_{\bar{a}}^0 = 1, \quad q_{\bar{b}}^0 = 1.$$

For agent 1, the set of available bundles is $\mathcal{X}^0 = \{ab, a, b, \emptyset\}$, from which she takes a. Then the capacities of objects will be updated as follows:

$$q_a^1 = q_a^0 - 1 = 0, \quad q_b^1 = q_b^0 = 1, \quad q_{\bar{a}}^1 = q_{\bar{a}}^0 = 1, \quad q_{\bar{b}}^1 = q_{\bar{b}}^0 - 1 = 0.$$

Hence the set of available bundles for agent 2 is $\mathcal{X}^1 = \{b\}$. This indicates that agent 2 has to take b and the final assignment is D below.

$$D = \begin{array}{c} \\ 1: \\ 2: \end{array} \begin{pmatrix} ab & a & b & \emptyset \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

■

With the above well-defined serial dictatorship rules, we define the random serial dictatorship rule as the equally weighted combination of these deterministic rules.

Definition 11. ***Random serial dictatorship for bundles (RSDB)*** *is a random assignment rule $RSDB : \mathbb{D}^n \to \mathcal{L}$ such that given a preference profile $P \in \mathbb{D}^n$,*

$$RSDB(P) = \frac{1}{|\Sigma|} \sum_{\sigma \in \Sigma} SDB^{\sigma}(P).$$

The performance of the RSDB rule on the universal domain is summarized below.

Proposition 2. *The RSDB rule on the universal domain satisfies decomposability, sd-strategy-proofness, equal treatment of equals but violates sd-efficiency.*

Proof. First, by construction the RSDB rule is decomposable, because it is a lottery over deterministic assignment rules. Second, the RSDB rule is sd-strategy-proof, because each SDB rule is sd-strategy-proof and that sd-strategy-proofness is preserved under linear combinations. Third, the RSDB rule treats equals equally, because the various orderings of agents are equally weighted.

The following example proves that the RSDB rule is not sd-efficient. This example is a modification of an example in Bogomolnaia and Moulin (2001).

Let $I = \{1, 2, 3, 4\}$, $X = \{a, b, c\}$, and $q_a = q_b = q_c = 1$. Consider the preference profile P given below. Then the random assignment specified by the RSDB rule is below, where B denotes an arbitrary bundle different from ab, c, and $\emptyset$. The reader can verify that it is not sd-efficient because it is dominated by the random assignment L.

$$\begin{array}{llllll} P_1, P_2: & ab & c & \emptyset & \cdots \\ P_3, P_4: & c & ab & \emptyset & \cdots \end{array}$$

$$RSDB(P) = \begin{array}{c} \\ 1,2: \\ 3,4: \end{array} \begin{pmatrix} ab & c & \emptyset & B \\ 5/12 & 1/12 & 1/2 & 0 \\ 1/12 & 5/12 & 1/2 & 0 \end{pmatrix} \quad L = \begin{array}{c} \\ 1,2: \\ 3,4: \end{array} \begin{pmatrix} ab & c & \emptyset & B \\ 1/2 & 0 & 1/2 & 0 \\ 0 & 1/2 & 1/2 & 0 \end{pmatrix}$$

■

4.2 The Probabilistic Serial Rule for Bundles

The PS rule in the classical random assignment model (Bogomolnaia and Moulin (2001)) is a special case of the so-called simultaneous eating algorithm, where all agents eat at the uniform speed. It treats the objects as if they are infinitely divisible and proceeds as follows: all the agents eat their respectively favorite objects at the uniform speed, until some object is exhausted; thereafter, agents eat their respectively favorite objects among the available ones, still at the uniform speed, until some other object is exhausted; this procedure is repeated until all the objects are exhausted. Finally, the share of an object eaten by an agent is interpreted as the probability that this agent gets this object.

The PS rule can be naturally extended to the setting with bundles, where every agent eats a bundle rather than an object. In particular, an agent eating a bundle means that she simultaneously eats every object contained in that bundle. As in the definition of the SDB rule, here too we introduce for each $x \in X$ an opposite object $\bar{x}$. An agent eating a bundle A is equivalent to saying that she eats every x such that $x \in A$ and every $\bar{x}$ such that $x \notin A$. We present the probabilistic serial rule for bundles below, where r_x^{v-1} and $r_{\bar{x}}^{v-1}$ denote respectively the available shares of x and $\bar{x}$ for the v-th step. Accordingly, $\mathcal{X}^{v-1}$ denotes the available bundles for the v-th step. In particular, the length of the v-th step, i.e., $t^v - t^{v-1}$, is defined as the shortest time needed to exhaust at least one in x's and $\bar{x}$'s.

Definition 12. ***Probabilistic serial rule for bundles (PSB)*** *is a random assignment rule PSB : $\mathbb{D}^n \to \mathcal{L}$ such that given a preference profile $P \in \mathbb{D}^n$, $PSB(P) \equiv L^{\bar{v}}$ where $L^{\bar{v}}$ is generated by the following algorithm.*

Let $t^0 = 0$, $\mathcal{X}^0 = \mathcal{X}$, $r_x^0 = q_x$ and $r_{\bar{x}}^0 = n - q_x$ for all $x \in X$.

Let in addition L^0 be a matrix of size $n \times |\mathcal{X}|$ with all zeros.

For $v = 1, \cdots, \bar{v}$,

$$I_x^v \equiv \{i \in I : x \in r_1(P_i, \mathcal{X}^{v-1})\}, \forall x \in X;$$

$$I_{\bar{x}}^v \equiv I \backslash I_x^v, \forall x \in X;$$

$$t^v \equiv t^{v-1} + \min\left\{\left\{\frac{r_x^{v-1}}{|I_x^v|} : r_x^{v-1} > 0\right\} \bigcup \left\{\frac{r_{\bar{x}}^{v-1}}{|I_{\bar{x}}^v|} : r_{\bar{x}}^{v-1} > 0\right\}\right\};$$

$$L_{iA}^v \equiv L_{iA}^{v-1} + \begin{cases} t^v - t^{v-1}, & \text{if } A = r_1(P_i, \mathcal{X}^{v-1}) \\ 0, & \text{otherwise} \end{cases}, \forall i \in I, A \in \mathcal{X}^{v-1};$$

$$r_x^v \equiv r_x^{v-1} - (t^v - t^{v-1}) \cdot |I_x^v|, \forall x \in X;$$

$$r_{\bar{x}}^v \equiv r_{\bar{x}}^{v-1} - (t^v - t^{v-1}) \cdot |I_{\bar{x}}^v|, \forall x \in X;$$

$$\mathcal{X}^v \equiv \mathcal{X}^{v-1} \backslash \{A \in \mathcal{X}^{v-1} : \exists x \in X \text{ s.t. } [x \in A, r_x^v = 0] \text{ or } [x \notin A, r_{\bar{x}}^v = 0]\};$$

where $\bar{v}$ is identified by $\mathcal{X}^{\bar{v}} = \emptyset$.

The following example illustrates an eating procedure.

Example 9. *Let $X = \{a, b, c\}$, $q_x = 1\ \forall x \in X$, and $I = \{1, 2, 3\}$. Let the preference profile P be as below.*

$$\begin{array}{llllll} P_1: & ab & abc & \cdots & \cdots \\ P_2: & \emptyset & ab & c & \cdots \\ P_3: & \emptyset & ab & c & \cdots \end{array} \qquad \begin{array}{lcccc} & \emptyset & ab & abc & c \\ L_1: & 0 & 2/3 & 1/3 & 0 \\ L_2: & 2/3 & 0 & 0 & 1/3 \\ L_3: & 2/3 & 0 & 0 & 1/3 \end{array}$$

Initially the available shares are $r^0_a = r^0_b = r^0_c = 1$ and $r^0_{\bar a} = r^0_{\bar b} = r^0_{\bar c} = 2$ and hence every bundle is available.

In the first period, agent 1 eats bundle ab and agents 2 and 3 eat $\emptyset$. So the sets of agents who eat various available objects are as follows:

$$\begin{array}{lll} I^1_a = \{1\} & I^1_b = \{1\} & I^1_c = \emptyset \\ I^1_{\bar a} = \{2,3\} & I^1_{\bar b} = \{2,3\} & I^1_{\bar c} = \{1,2,3\}. \end{array}$$

The object $\bar c$ will be exhausted first since $r^0_{\bar c}/|I^1_{\bar c}| = 2/3$ is the smallest among available objects. This also identifies the end of the first period, i.e., $t^1 = 2/3$. Hence in the first period agent 1 eats $2/3$ of ab and agents 2 and 3 each eats $2/3$ of $\emptyset$. We now update the available shares of objects as below

$$\begin{array}{ll} r^1_a = r^0_a - 2/3 \cdot |I^1_a| = 1/3 & r^1_{\bar a} = r^0_{\bar a} - 2/3 \cdot |I^1_{\bar a}| = 2/3 \\ r^1_b = r^0_b - 2/3 \cdot |I^1_b| = 1/3 & r^1_{\bar b} = r^0_{\bar b} - 2/3 \cdot |I^1_{\bar b}| = 2/3 \\ r^1_c = r^0_c - 2/3 \cdot |I^1_c| = 1 & r^1_{\bar c} = r^0_{\bar c} - 2/3 \cdot |I^1_{\bar c}| = 0. \end{array}$$

So, except for $\bar c$, all the other objects are still available, which defines the available bundles as $\mathcal{X}^1 = \{abc, ac, bc, c\}$. In particular, the set $\emptyset$ is not available any more.

In the second period, agent 1 eats abc, which is her favorite in $\mathcal{X}^1$ and agents 2 and 3 eat c, which is their favorite in $\mathcal{X}^1$. The sets of agents who eat various available objects are as follows:

$$\begin{array}{lll} I^2_a = \{1\} & I^2_b = \{1\} & I^2_c = \{1,2,3\} \\ I^2_{\bar a} = \{2,3\} & I^2_{\bar b} = \{2,3\} & I^2_{\bar c} = \emptyset. \end{array}$$

Then all the objects will be exhausted at the same time $t^2 = t^1 + 1/3 = 1$ since $1/3 = r^1_a/|I^2_a| = r^1_b/|I^2_b| = r^1_c/|I^2_c| = r^1_{\bar a}/|I^2_{\bar a}| = r^1_{\bar b}/|I^2_{\bar b}|$. In this period agent 1 eats $1/3$ of abc and each of agents 2 and 3 eats $1/3$ of c. At the end of the second period, all the objects are exhausted and hence the algorithm terminates. Hence the resulting random assignment is L presented above. ■

As the above example indicates, the PSB rule is well-defined. We proceed by presenting the following example to show that this rule is not sd-strategy-proof on the universal domain.

Example 10. *Let $I = \{1, 2, 3\}$, $X = \{a, b\}$, and $q_a = q_b = 1$. Two preferences are as below.*

$$\begin{array}{lccccccc} \tilde P_i: & ab & \succ & a & \succ & b & \succ & \emptyset \\ \hat P_i: & a & \succ & ab & \succ & b & \succ & \emptyset \end{array}$$

Let two preference profiles be $P = (\tilde{P}_1, \tilde{P}_2, \hat{P}_3, \hat{P}_4)$ and $P' = (\hat{P}_1, \tilde{P}_2, \hat{P}_3, \hat{P}_4)$. The following are the corresponding assignments specified by the PSB rule. In particular, $L = PSB(P)$ and $L' = PSB(P')$.

$$
\begin{array}{rcccc}
 & ab & a & b & \emptyset \\
L_1: & 1/4 & 0 & 1/8 & 5/8 \\
L_2: & 1/4 & 0 & 1/8 & 5/8 \\
L_3: & 0 & 1/4 & 1/8 & 5/8 \\
L_4: & 0 & 1/4 & 1/8 & 5/8
\end{array}
\qquad
\begin{array}{rcccc}
 & ab & a & b & \emptyset \\
L'_1: & 0 & 1/4 & 3/16 & 9/16 \\
L'_2: & 1/4 & 0 & 3/16 & 9/16 \\
L'_3: & 0 & 1/4 & 3/16 & 9/16 \\
L'_4: & 0 & 1/4 & 3/16 & 9/16
\end{array}
$$

Across the two preference profiles, agent 1 is the unique deviator. Notice that $L'_{1ab} + L'_{1a} + L'_{1b} = 7/16 > 6/16 = L_{1ab} + L_{1a} + L_{1b}$, which means that by misreporting P'_1, agent 1 receives a higher probability of getting a bundle better than $\emptyset$. Hence the PSB rule is manipulable on any domain containing these two preferences, including the universal domain. ■

The performance of this rule on the universal domain is summarized below.

Proposition 3. *The PSB rule on the universal domain satisfies sd-efficiency and equal treatment of equals but violates sd-strategy-proofness and decomposability.*

Proof. The previous example shows that the PSB rule is not sd-strategy-proof on the universal domain. We use Example 15 in Section 5 to demonstrate that the PSB rule is not decomposable. The fact that the PSB rule treats equals equally follows from the fact that all the agents have the same eating speed at every point in time.

Now we turn to proving the efficiency claim. Let $P \in \mathbb{P}^n$ and $L = PSB(P)$. We show that L is sd-efficient at P. To do so, $\forall\, x \in X$, let $t(x)$ be the time when x is depleted, i.e., $t(x) \equiv \min\{t^v : r^v_x \leqslant 0\}$. Similarly, $\forall\, x \in X$, let $t(\bar{x})$ be the time when $\bar{x}$ is depleted, i.e., $t(\bar{x}) \equiv \min\{t^v : r^v_{\bar{x}} \leqslant 0\}$. We consider two cases.

Case 1: $\forall x \in X$, $t(x) \leqslant t(\bar{x})$. Lemma 1 below considers this case.

Lemma 1. *Let $P \in \mathbb{P}^n$ and $L = PSB(P)$. If, $\forall x \in X$, $t(x) \leqslant t(\bar{x})$, L is sd-efficient at P.*

Proof. We prove the lemma by contradiction. Suppose L is not sd-efficient at P. Then by Proposition 1, L is strongly balanced at P. Put otherwise, there is a subset $\mathcal{S} \subset \mathcal{T}$ such that (i) $(i, A, B) \in \mathcal{S}$ implies $B\ P_i\ A$ and $L_{iA} > 0$; (ii) $\forall x \in X$, $[\exists (i, A, B) \in \mathcal{S}$ s.t. $x \in A \backslash B] \Leftrightarrow [\exists (i, A, B) \in \mathcal{S}$ s.t. $x \in B \backslash A]$. For each $i \in I$ and $A \in \mathcal{X}$ such that $L_{iA} > 0$, let $t(i, A)$ denote the time when agent i starts to consume A. Formally, $t(i, A) \equiv \min\{t(x) : x \in A\} - L_{iA}$.

Pick an arbitrary $(i_1, A_1, B_1) \in \mathcal{S}$, by definition, $B_1\ P_{i_1}\ A_1$ and $L_{i_1 A_1} > 0$. Hence at the time when agent i_1 starts to consume A_1, i.e., $t(i_1, A_1)$, B_1 is already depleted. Then the assumption that, $\forall\, x \in X$, $t(x) \leqslant t(\bar{x})$, implies the existence of $x_1 \in B_1 \backslash A_1$ such that $t(x_1) \leqslant t(i_1, A_1)$. Strong balancedness then implies the existence of $(i_2, A_2, B_2) \in \mathcal{S}$ such that $x_1 \in A_2 \backslash B_2$. Then $L_{i_2 A_2} > 0$ implies $t(i_2, A_2) < t(x_1)$. Similarly, let $x_2 \in B_2 \backslash A_2$ be arbitrary such that $t(x_2) \leqslant t(i_2, A_2)$. Hence $t(x_2) \leqslant t(i_2, A_2) < t(x_1)$. We repeat the procedure to find x_3, A_3, and i_3 such that $t(x_3) \leqslant t(i_3, A_3) < t(x_2) \leqslant t(i_2, A_2) < t(x_1)$. If $x_3 = x_1$, we have a contradiction. Otherwise, we repeat the procedure to find x_4, and so on. Finally, the finiteness of X implies the existence of x such that $t(x) < t(x)$: contradiction. ■

Case 2: Let $\bar{X} \equiv \{x \in X : t(x) > t(\bar{x})\}$ be nonempty. Let $\mathcal{E} \equiv (I, X, q)$ denote the model setting. We define a new model $\mathcal{E}' \equiv (I, Y, p)$ such that (i) the set of agents is the same as the original model, and (ii) the set of objects Y and their capacities p are associated to X and q via an arbitrary bijection $f : Y \to X$, as follows.

$$p_y = \begin{cases} q_{f(y)}, & f(y) \in X \backslash \bar{X} \\ n - q_{f(y)}, & f(y) \in \bar{X} \end{cases}$$

Thus if an object $y \in Y$ is mapped to an object x not in $\bar{X}$, its capacity is the same as x. Otherwise, its capacity is defined as n minus the capacity of x. For $\mathcal{E}$ and $\mathcal{E}'$, the set of bundles are denoted as $\mathcal{X}$ and $\mathcal{Y}$ respectively. It is evident that $|\mathcal{X}| = |\mathcal{Y}|$. In addition, the set of random assignments are denoted as $\mathcal{L}$ and $\mathcal{L}'$ respectively. We now define a mapping $g : \mathcal{Y} \to \mathcal{X}$ such that $\forall\ A \in \mathcal{Y}$, $g(A) = B \in \mathcal{X}$ if and only if, $\forall y \in Y$,

$$\begin{aligned} &f(y) \in B, \text{ if either } [f(y) \in X\backslash\bar{X} \text{ and } y \in A] \text{ or } [f(y) \in \bar{X} \text{ and } y \notin A] \\ &f(y) \notin B, \text{ otherwise.} \end{aligned} \tag{1}$$

For a better understanding of the construction, we illustrate in Example 11 the construction with a specific model setting. One can verify that g is a bijection. For the new model $\mathcal{E}'$, we specify a profile of preferences on $\mathcal{Y}$, denoted as $P' = (P'_i)_{i \in I}$, and a random assignment $L' \in \mathcal{L}'$. In particular, for all $i \in I$ and $A, B \in \mathcal{Y}$, $A\ P'_i\ B$ if and only if $g(A)\ P_i\ g(B)$. For all $i \in I$ and $A \in \mathcal{Y}$, $L'_{iA} = L_{ig(A)}$.

We now make the following two claims.

Claim 1: L' is sd-efficient at P' in $\mathcal{E}' \Rightarrow L$ is sd-efficient at P in $\mathcal{E}$.

We prove the contrapositive statement. Let L be not sd-efficient at P in $\mathcal{E}$. Then, by definition, $\exists\ \tilde{L} \in \mathcal{L}$ such that $\tilde{L} \neq L$ and $\tilde{L}_i\ P_i^{sd}\ L_i$ for all $i \in I$. We construct a matrix $\tilde{L}' \in [0,1]^{I \times \mathcal{Y}}$ such that, $\forall\ i \in I$ and $A \in \mathcal{Y}$, $\tilde{L}'_{iA} = \tilde{L}_{ig(A)}$. We prove the following three statements.

1. $\tilde{L}'$ is a random assignment in $\mathcal{E}'$, i.e., $\tilde{L}' \in \mathcal{L}'$.

To see this, note that, $\forall i \in I$,

$$\sum_{A \in \mathcal{Y}} \tilde{L}'_{iA} = \sum_{A \in \mathcal{Y}} \tilde{L}_{ig(A)} = \sum_{A \in \mathcal{X}} \tilde{L}_{iA} = 1.$$

Moreover, $\forall\ y \in Y$,

$$\begin{aligned} f(y) \in X\backslash\bar{X} : \quad \sum_{i \in I, y \in A} \tilde{L}'_{iA} &= \sum_{i \in I, y \in A} \tilde{L}_{ig(A)} && (\text{ by } \tilde{L}'_{iA} = \tilde{L}_{ig(A)}) \\ &= \sum_{i \in I, f(y) \in A} \tilde{L}_{iA} && (\text{ by the definition of } g) \\ &= q_{f(y)} && (\text{ by} \tilde{L} \in \mathcal{L}) \\ &= p_y. && (\text{ by the definition of } p) \end{aligned}$$

$$
\begin{aligned}
f(y) \in \bar{X}: \quad \sum_{i\in I, y\in A} \tilde{L}'_{iA} &= \sum_{i\in I, y\in A} \tilde{L}_{ig(A)} && (\text{ by } \tilde{L}'_{iA} = \tilde{L}_{ig(A)})\\
&= \sum_{i\in I, f(y)\notin A} \tilde{L}_{iA} && (\text{ by the definition of } g)\\
&= n - \sum_{i\in I, f(y)\in A} \tilde{L}_{iA} = n - q_{f(y)} && (\text{ by} \tilde{L} \in \mathcal{L})\\
&= p_y. && (\text{ by the definition of } p)
\end{aligned}
$$

2. $\tilde{L}' \neq L'$. This is implied by the fact that $\tilde{L} \neq L$ and that g is a bijection.
3. $\forall\, i \in I$, $\tilde{L}'_i\ P'^{sd}_i\ L'_i$. Given $\forall\, i \in I$, $B \in \mathcal{Y}$, and the fact that $\tilde{L}_i\ P^{sd}_i\ L_i$,

$$
\sum_{\{A\in\mathcal{Y}:AR'_iB\}} \tilde{L}'_{iA} - \sum_{\{A\in\mathcal{Y}:AR'_iB\}} L'_{iA} = \sum_{\{g(A)\in\mathcal{X}:g(A)R_ig(B)\}} \tilde{L}_{iA} - \sum_{\{g(A)\in\mathcal{X}:g(A)R_ig(B)\}} L_{iA} \geqslant 0.
$$

The above three statements together imply that L' is not sd-efficient at P' in $\mathcal{E}'$.

Claim 2: For $\mathcal{E}'$, $L' = PSB(P')$ and $t(y) \leqslant t(\bar{y})$ for all $y \in Y$.

By construction, when the PSB rule is applied to P' in $\mathcal{E}'$, if $f(y) \in X\backslash\bar{X}$, y mimics $f(y)$ when the PSB rule is applied to P in $\mathcal{E}$. If instead $f(y) \in \bar{X}$, $\bar{y}$ mimics $f(y)$ when the PSB rule is applied to P in $\mathcal{E}$. Hence, $\forall\, i \in I$, $y \in Y$ such that $f(y) \in X\backslash\bar{X}$, and any point in time, agent i consumes y when the PSB rule is applied to P' in $\mathcal{E}'$ if and only if agent i consumes $f(y)$ when the PSB rule is applied to P in $\mathcal{E}$. So $t'(y) = t(x) \leqslant t(\bar{x}) = t'(\bar{y})$, where $t'(y)$ denote the point in time when object y is depleted and $x \in X$ such that $f(y) = x$. On the contrary, $\forall$ $i \in I$, $y \in Y$ such that $f(y) \in \bar{X}$, and any point in time, agent i consumes y when the PSB rule is applied to P' in $\mathcal{E}'$ if and only if agent i consumes $\bar{x}$ when the PSB rule is applied to P in $\mathcal{E}$, where $x = f(y)$. So $t'(y) = t(\bar{x}) \leqslant t(x) = t'(\bar{y})$, where $x \in X$ such that $f(y) = x$.

The statement that $L = PSB(P)$ is sd-efficient at P for Case 2 is now implied by Claims 1, 2 and Lemma 1. In particular, Claim 2 and Lemma 1 imply that L' is sd-efficient at P' in $\mathcal{E}'$. Then Claim 1 implies that L is sd-efficient at P in $\mathcal{E}$. ■

Example 11. *Consider the setting of Example 9. In particular, the model is denoted as* $\mathcal{E} = (I, X, q)$, *where* $I = \{1, 2, 3\}$, $X = \{a, b, c\}$, *and* $q_a = q_b = q_c = 1$. *According to the eating procedure illustrated in Example 9,* $t(a) = t(\bar{a}) = 1$, $t(b) = t(\bar{b}) = 1$, $t(\bar{c}) = 2/3 < 1 = t(c)$. *Hence, let* $\bar{X} = \{c\}$. *Let in addition,* $\mathcal{E}' = (I, Y, p)$ *be the new model where* $Y = \{x, y, z\}$, $p_x = q_a = 1$, $p_y = q_b = 1$, *and* $p_z = 3 - q_c = 2$. *Let* $f : \{x, y, z\} \to \{a, b, c\}$ *be such that* $f(x) = a$, $f(y) = b$, *and* $f(z) = c$. *Given this, we map the bundles in* $\mathcal{E}'$ *to the ones in* $\mathcal{E}$ *according to the definition of* g *in (1).*

$$
\begin{array}{cccccccccc}
 & xy & xyz & x & y & xz & yz & \emptyset & z\\
g: & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow\\
 & abc & ab & ac & bc & a & b & c & \emptyset
\end{array}
$$

Given the preference profile P and the PSB assignment L in $\mathcal{E}$, we construct the profile P' and a matrix L' in $\mathcal{E}'$ via g.

$$\begin{array}{lcccc} P'_1: & xyz & xy & \cdots & \cdots \\ P'_2: & z & xyz & \emptyset & \cdots \\ P'_3: & z & xyz & \emptyset & \cdots \end{array} \qquad \begin{array}{lcccc} & z & xyz & xy & \emptyset \\ L'_1: & 0 & 2/3 & 1/3 & 0 \\ L'_2: & 2/3 & 0 & 0 & 1/3 \\ L'_3: & 2/3 & 0 & 0 & 1/3 \end{array}$$

One can easily verify that L' is exactly the random assignment generated by applying the PSB rule to P'. One can also verify that in the corresponding eating procedure, everyone of the objects, x, y, and z in particular, is depleted before its opposite object. ■

In the remainder of this subsection, we illustrate an implication of our feasibility requirement on designing a desirable random assignment rule: a possibility result in the setting with objects fails in the setting with bundles. In the setting with objects, Liu (2017) proved that the PS rule is sd-strategy-proof on the sequentially dichotomous domain, which is generated by lexicographically checking a fixed list of properties. We first introduce the sequentially dichotomous domain to the current setting.

Let $x_1, x_2, \cdots, x_m$ be a fixed ordering of objects. To simplify notation, we denote a bundle as a 0-1 vector of length m, where a "one" at the t-th position means this bundle contains x_t and a "zero" means not. For example $A = (1, 0, 0, 1)$ is equivalent to $A = x_1x_4$. For each bundle A and each index $t = 1, \cdots, m$, we write A_t as its t-th element and A^t the sequence of the first t elements. For example, for the bundle $A = (1, 0, 0, 1)$, $A_2 = 0$ and $A^3 = (1, 0, 0)$. A preference is sequentially dichotomous if the bundles are ranked in the following sequential way. First, either every bundle containing x_1 is better than every bundle that does not, or the other way around. Next, within the bundles containing x_1, either every bundle containing x_2 is better than every bundle that does not, or the other way around. Similarly, within the bundles that do not contain x_1, either every one containing x_2 is better than every one that does not, or the other way around. The preference is refined by checking sequentially for x_3, x_4, and so on.

Formally, a preference $P_i \in \mathbb{P}$ is sequentially dichotomous if

1. either $[\forall A, B \in \mathcal{X}$ s.t. $A_1 = 1, B_1 = 0, A\ P_i\ B]$ or $[\forall A, B \in \mathcal{X}$ s.t. $A_1 = 1, B_1 = 0, B\ P_i\ A]$;

2. $\forall t = 2, \cdots, m$ and $\forall \alpha \in \{0, 1\}^{t-1}$, either $[\forall A, B \in \mathcal{X}$ s.t. $A^{t-1} = B^{t-1} = \alpha, A_t = 1, B_t = 0, A\ P_i\ B]$ or $[\forall A, B \in \mathcal{X}$ s.t. $A^{t-1} = B^{t-1} = \alpha, A_t = 1, B_t = 0, B\ P_i\ A]$.

In this manner, the preference structure of the sequentially dichotomous domain of Liu (2017) is directly introduced into the bundle setting. One can find that, by treating "containing x_t" as the t-th property, the above given definition is equivalent to the original one. The preferences $\tilde{P}_i$ and $\hat{P}_i$ in Example 10 are instances of sequentially dichotomous preferences. In particular, agents compare the bundles by checking first whether the bundle contains a and second whether it contains b. Both preferences prefer the bundles containing a (ab and a) to the bundles that do not (b and $\emptyset$). Then between ab and a, $\tilde{P}_i$ prefers the one containing b while $\hat{P}_i$ prefers the one that does not. Between b and $\emptyset$, both preferences prefer the one containing b.

Hence the manipulation of the PSB rule in Example 10 indicates that the possibility result on the sequentially dichotomous domain in the classical random assignment model fails in the setting with bundles. This failure occurs exactly because the definition of feasibility is modified. If we treat the four bundles as distinct objects, feasibility of random assignments of individual objects now dictates that every column sums to one. Then the random assignments generated by the PS rule for the above profiles would be as follows.

$$
\begin{array}{ccccc}
 & ab & a & b & \emptyset \\
L_1: & 1/2 & 0 & 1/4 & 1/4 \\
L_2: & 1/2 & 0 & 1/4 & 1/4 \\
L_3: & 0 & 1/2 & 1/4 & 1/4 \\
L_4: & 0 & 1/2 & 1/4 & 1/4
\end{array}
\qquad
\begin{array}{ccccc}
 & ab & a & b & \emptyset \\
L'_1: & 1/6 & 1/3 & 1/4 & 1/4 \\
L'_2: & 1/2 & 0 & 1/4 & 1/4 \\
L'_3: & 1/6 & 1/3 & 1/4 & 1/4 \\
L'_4: & 1/6 & 1/3 & 1/4 & 1/4
\end{array}
$$

It is evident that agent 1's misreport is no longer profitable. To summarize, it follows that the change in the feasibility requirement from the classical setting to the setting with bundles has a significant implication on possibilities of designing a desirable rule, in that a previously known possibility result fails.

4.3 The Essentially Monotonic Preferences

In order to introduce our preference restriction, we identify first the following sequence of bundles and integers. We call them **critical bundles** and **critical capacities**.

$$
\begin{aligned}
&A_1 \equiv X, && d_1 \equiv \min\{q_x : x \in X\}, \\
&A_2 \equiv \{x \in X : q_x > d_1\}, && d_2 \equiv \min\{q_x - d_1 : q_x > d_1\}, \\
&\vdots && \vdots \\
&A_k \equiv \left\{x \in X : q_x > \textstyle\sum_{l=1}^{k-1} d_l\right\}, && d_k \equiv \min\left\{q_x - \textstyle\sum_{l=1}^{k-1} d_l : q_x > \textstyle\sum_{l=1}^{k-1} d_l\right\}, \\
&\vdots && \vdots \\
&A_{K-1} \equiv \left\{x \in X : q_x > \textstyle\sum_{l=1}^{K-2} d_l\right\}, && d_{K-1} \equiv \min\left\{q_x - \textstyle\sum_{l=1}^{K-2} d_l : q_x > \textstyle\sum_{l=1}^{K-2} d_l\right\}, \\
&A_K \equiv \left\{x \in X : q_x > \textstyle\sum_{l=1}^{K-1} d_l\right\}, && d_K \equiv n - \textstyle\sum_{l=1}^{K-1} d_l,
\end{aligned}
$$

where K is identified by $A_K = \emptyset$. It is evident that $d_{K-1} = \max\{q_x : x \in X\}$. By the structure above, $X = A_1 \supsetneqq A_2 \supsetneqq \cdots \supsetneqq A_{K-1} \supsetneqq A_K = \emptyset$.

Example 12. Consider the situations where objects have the same number of copies, i.e., $q_x = q_y$ for all $x, y \in X$, the corresponding critical bundles and capacities are as below.

$$
\begin{array}{ll}
A_1 = X & d_1 = q_x \\
A_2 = \emptyset & d_2 = n - q_x
\end{array}
$$

■

Example 13. Consider a situation where $n = 6$, $X = \{a, b, c\}$, $q_a = 4$, $q_b = 3$, and $q_c = 2$. Then the critical bundles and capacities are as follows.

$$\begin{array}{ll} A_1 = abc & d_1 = 2 \\ A_2 = ab & d_2 = 1 \\ A_3 = a & d_3 = 1 \\ A_4 = \emptyset & d_4 = 2. \end{array}$$

■

A preference is called essentially monotonic if whenever a bundle is a proper subset of a critical bundle, it is less preferred to this critical bundle. Formally

Definition 13. *A preference $P_i \in \mathbb{P}$ is **essentially monotonic** if for any critical bundle A_k and any $A \in \mathcal{X}$ such that $A \subsetneqq A_k$, $A_k \ P_i \ A$.*

Let $\mathbb{D}_{EM} \subset \mathbb{P}$ be the set of all essentially monotonic preferences and call it **the essentially monotonic domain**. As shown by Examples 12 and 13, the more capacities vary, the more critical bundles will be identified. Hence more restrictions will be imposed on essentially monotonic preferences and $\mathbb{D}_{EM}$ will be smaller.

Among the preference restrictions studied in the setting with bundles, two of them are closely related to essentially monotonicity: **monotonicity** (Pápai, 2000b) and **separability** (Le Breton and Sen, 1999). A preference is monotonic if whenever a bundle is a proper subset of another bundle, the former is less preferred than the later. Formally, $\forall A, B \in \mathcal{X}$, $B \subsetneqq A \Rightarrow A \ P_i \ B$. A preference is separable if adding an additional object to a bundle is preferred if and only if the object itself is preferred to the empty bundle. Formally, $\forall A \in \mathcal{X}$ and $x \in X \backslash A$, $A \cup \{x\} \ P_i \ A$ if and only if $x \ P_i \ \emptyset$.

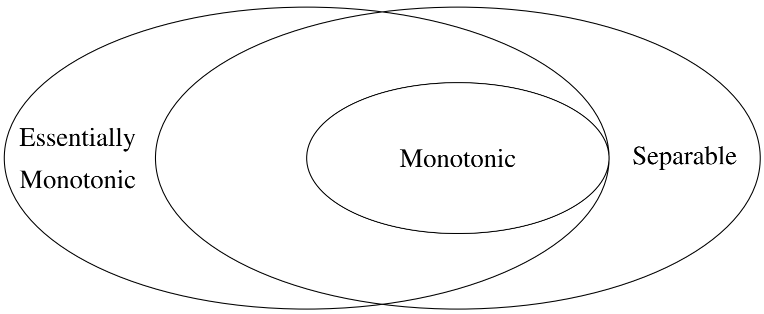


Figure 7: The Relationship among Preference Restrictions

Figure 7 shows the relationship. By definition, essential monotonicity is strictly weaker than monotonicity because the requirement that if a bundle is contained in a critical bundle it is less preferred is imposed only for critical bundles. This relation is true independent of the capacities. (Recall that the size of the essentially monotonic domain varies with the capacities.) Consider the critical bundles in Example 12, essential monotonicity requires only that the whole bundle, X, is the top ranked bundle. But monotonicity requires much more than that and hence the monotonic domain contains much fewer preferences than the essentially monotonic domain. For another instance, consider the critical bundles in Example 13, essential monotonicity requires (i) the whole set X is top ranked, (ii) bundles a, b, and $\emptyset$ are less preferred than ab, and (iii) $\emptyset$ is less preferred than a. Hence essential monotonicity imposes less structure on preferences than does monotonicity.

The essentially monotonic domain and the separable domain overlap with each other but no one contains the other. This relation is also true independent of the capacities. Consider Example 13 where there are totally 3 objects and 4 critical bundles. Recall that given the object set, these capacities identify a maximal set of critical bundles. In other words, the resulting essentially monotonic domain is minimal. Even in this case, there still exists a preference, P_i below, which is essentially monotonic but not separable. The critical bundles are underlined, from which essential monotonicity can be verified. To see that P_i is non-separable, notice that $\emptyset \ P_i \ c$ but $ac \ P_i \ a$. In addition P_i' below is a preference which is separable but not essentially monotonic.

$$P_i: \quad \underline{abc} \succ \underline{ab} \succ ac \succ bc \succ \underline{a} \succ b \succ \underline{\emptyset} \succ c$$

$$P_i': \quad \underline{\emptyset} \succ \underline{a} \succ b \succ c \succ \underline{ab} \succ ac \succ bc \succ \underline{abc}$$

4.4 The Equivalence

This subsection shows that the RSDB rule and the PSB rule select the same random assignment for any arbitrary profile of essentially monotonic preferences.

We present an example below.

Example 14. *Consider the setting of Example 13. Assume that all six agents have the same preference below, where the critical bundles are underlined.*

$$P_i: \quad \underline{abc} \quad c \quad ac \quad \underline{ab} \quad b \quad \underline{a} \quad \underline{\emptyset} \quad bc.$$

The reader can verify that both the RSDB rule and the PSB rule specify a random assignment where all agents have the same lottery as follows.

$$\begin{array}{lcccccccc} & abc & c & ac & ab & b & a & \emptyset & bc \\ L_i: & 1/3 & 0 & 0 & 1/6 & 0 & 1/6 & 1/3 & 0 \end{array}$$

■

Remark 1. Recall that, in order to fully allocate the unwanted objects, we introduced an opposite object $\bar{x}$ for each $x \in X$. In addition, in the definition of the RSDB rule and the PSB rule, one needs to keep track of the availability of every x and every $\bar{x}$. As illustrated by Example 7, this is because, in general, there could be "goods" (a in Example 7) and "bads" (b in Example 7) at the same time. However, under essential monotonicity, objects are either all goods or all bads. In the case where they are all goods, there is no need to keep track of the availability of opposite objects. This can be seen when one verifies the random assignment in Example 14 as the result of both the RSDB rule and the PSB rule. In the cases where objects are all bads, one can easily construct an imaginary problem, as shown in Subsection 1.1, and hence the analogous logic applies.

It turns out that the above example is not a coincidence but true for all profiles of essentially monotonic preferences.

Theorem 2. *The RSDB rule is equivalent to the PSB rule on the essentially monotonic domain.*

Proof. Let $P \in \mathbb{D}_{EM}^n$ be an arbitrary profile of essentially monotonic preferences. We show $\forall i \in I$, $RSDB_i(P) = PSB_i(P) = L_i$ as below, where B denotes a bundle not included in the critical bundles $A_1, \cdots, A_K$.

$$\begin{array}{cccccccc} & A_1 & A_2 & \cdots & A_{K-1} & A_K & B \\ L_i: & \frac{d_1}{n} & \frac{d_2}{n} & \cdots & \frac{d_{K-1}}{n} & \frac{d_K}{n} & 0 \end{array}$$

The fact that $PSB_i(P) = L_i$ is seen as follows: from the beginning all agents eat their commonly favorite bundle, X, which will be exhausted at d_1/n. Thereafter, within the available bundles, essential monotonicity implies that the favorite bundle of agents is commonly A_2. So all agent start to eat A_2, which will then be exhausted at $d_1/n + d_2/n$. The iteration goes on until $d_1/n + \cdots + d_{K-1}/n$, when all the objects are depleted. Then every agent receives $1 - (d_1/n + \cdots + d_{K-1}/n)$ probability on $\emptyset$, which is equivalent to specifying for each agent i $L_{i\emptyset} = d_K/n$.

To see $RSDB_i(P) = L_i$, notice that given an arbitrary ordering σ of agents, $SDB^{\sigma}_{\sigma(i)}(P) = X$ for all $i = 1, \cdots, d_1$. In other words, each of the first d_1 agents will take their commonly favorite bundle X. Thereafter, the agents ordered from $d_1 + 1$ to $d_1 + d_2$ will take A_2 since essential monotonicity implies that this is their favorite bundle in the available ones. This argument continues and gives that $SDB^{\sigma}_{\sigma(i)}(P) = A_k$ for all $i = d_1 + \cdots + d_{k-1} + 1, d_1 + \cdots + d_{k-1} + 2, \cdots, d_1 + \cdots + d_{k-1} + d_k$ and all $k = 2, \cdots, K$. As the equally weighted average of all serial dictatorship rules, for an arbitrary agent, there is probability $1/n$ to be ordered on each of n positions: There are a total of $n!$ orderings of agents, among which there are $(n-1)!$ orderings where agent i is ordered at the k-th position. Hence $RSDB_i(P) = L_i$.

Note that the random assignment L is determined by the critical bundles and critical capacities. In particular, it is independent of the preferences in P. Hence, we have proved that these two rules degenerate to a constant rule on the essentially monotonic domain. ■

The above equivalence gives the following result on the existence of a desirable rule.

Corollary 1. *There is a decomposable random assignment rule on the essentially monotonic domain satisfying sd-efficiency, sd-strategy-proofness, and equal treatment of equals.*

Remark 2. A by-product of Theorem 2 is that the PSB rule and the RSDB rule both degenerate to a constant rule. In other words, given a specific problem, we simply identify the critical bundles and then allocate equally these bundles, regardless of the preferences of agents. As mentioned in the introduction, the fact that sd-efficiency is preserved by such a simple constant assignment on such a large domain comes as a surprise. ■

4.5 Impossibility

This subsection presents an impossibility on the universal domain. Recall that the capacities of objects are collected in a vector $q = (q_x)_{x \in X}$. Given q, we identify the critical bundles

$A_1, \cdots, A_K$ and in particular denote K as the number of critical bundles. Since the whole bundle X and the empty bundle $\emptyset$ are identified as critical bundles no matter what the capacity vector q is, $K \geqslant 2$. In principle, the more capacities vary, the larger K is. Below, we present a general impossibility, which states that when there are at least four critical bundles, no rule on the universal domain satisfies sd-strategy-proofness, sd-efficiency, and sd-envy-freeness at the same time. Sd-envy-freeness is a fairness axiom stronger than equal treatment of equals and requires that an agent always weakly prefers her own lottery to any other's. Formally, a rule $\varphi : \mathbb{D}^n \to \mathcal{L}$ is **sd-envy-free** if $\forall P \in \mathbb{D}^n$ and $i, j \in I$, $\varphi_i(P)\ P_i^{sd}\ \varphi_j(P)$.

Proposition 4. *Given $K \geqslant 4$. There is no sd-strategy-proof, sd-efficient, and sd-envy-free rule on the universal domain.*

Proof. Recall that the number of critical bundles is at most $1 + \max\{q_x : x \in X\}$. In addition, we require $q_x \leqslant n - 1$ for all $x \in X$. Hence, by construction, $n \geqslant K \geqslant 4$. Let $\varphi : \mathbb{P}^n \to \mathcal{L}$ be an sd-strategy-proof, sd-efficient, and sd-envy-free rule. In the following, we construct four preference profiles P^1, P^2, P^3, P^4, and then characterize the random assignments $\varphi(P^1)$, $\varphi(P^2)$, $\varphi(P^3)$, $\varphi(P^4)$. Finally, a contradiction against feasibility is identified, which then proves the theorem.

The preference profiles we construct consist of only the following three preferences

$$\begin{array}{lccccccc}
\bar{P}_i: & A_1 & A_3 & A_2 & A_4 & \cdots & A_K & \cdots \\
P_i: & A_1 & A_2 & A_3 & A_4 & \cdots & A_K & \cdots \\
\hat{P}_i: & A_2 & A_1 & A_3 & A_4 & \cdots & A_K & \cdots
\end{array}$$

Specifically, the critical bundles are top-ranked and the ranking of A_4 through A_K is the same across the three preferences.

Claim 1: Let the first preference profile be such that all the agents have the same preference as P_i in above table, i.e., $P^1 = (P_1, P_2, P_3, \cdots, P_n)$. Then $\varphi(P^1)$ is as below.

$$\begin{array}{rccccccc}
 & A_1 & A_2 & A_3 & A_4 & \cdots & A_K & \cdots \\
1 \cdots n: & \frac{d_1}{n} & \frac{d_2}{n} & \frac{d_3}{n} & \frac{d_4}{n} & \cdots & \frac{d_K}{n} & 0
\end{array}$$

By sd-envy-freeness, agents have the same lottery. In order to verify the claim, we notice that these exists $x \in A_1$ such that $q_x = d_1$. Then feasibility of x together with sd-envy-freeness require $L_{iA_1} \leqslant \frac{d_1}{n}$. Similarly, there exists $x \in A_1 \cap A_2$ such that $q_x = d_1 + d_2$, which then implies $L_{iA_1} + L_{iA_2} \leqslant \frac{d_1}{n} + \frac{d_2}{n}$. This argument proceeds until $\sum_{k=1}^{K} L_{iA_k} \leqslant 1$. It is hence evident that any sd-envy-free and feasible assignment $L \neq \varphi(P^1)$ is dominated by $\varphi(P^1)$.

Claim 2: Let $P^2 = (\bar{P}_1, P_2, P_3, \cdots, P_n)$. Then $\varphi(P^2)$ is as follows.

$$\begin{array}{rccccccc}
 & A_1 & A_2 & A_3 & A_4 & \cdots & A_K & \cdots \\
1: & \frac{d_1}{n} & 0 & \frac{d_2+d_3}{n} & \frac{d_4}{n} & \cdots & \frac{d_K}{n} & 0 \\
2 \cdots n: & \frac{d_1}{n} & \frac{d_2}{n-1} & \frac{d_3 - \frac{d_2+d_3}{n}}{n-1} & \frac{d_4}{n} & \cdots & \frac{d_K}{n} & 0
\end{array}$$

From P^1 to P^2, agent 1 is the unilateral deviator and she reversed the ranking of A_2 and A_3 with no other change. Hence sd-strategy-proofness implies $\varphi_{1A}(P^2) = \varphi_{1A}(P^1)$ for all

$A \neq A_2, A_3$. Then sd-envy-freeness implies $\varphi_{iA}(P^2) = \varphi_{1A}(P^2)$ for all $i = 2, \cdots, n$ and all $A \neq A_2, A_3$. Notice that sd-efficiency implies $\varphi_{1A_2}(P^2) = 0$ since otherwise, $\varphi_{iA_3}(P^2) = 0$ for all $i = 2, \cdots, n$, which implies $\varphi_{1A_3}(P^2) = d_3$ and hence $\varphi_{1A_2}(P^2) + \varphi_{1A_3}(P^2) > d_3 \geqslant 1$: contradiction to feasibility. Given $\varphi_{1A_2}(P^2) = 0$, $\varphi_{1A_3}(P^2) = \frac{d_2+d_3}{n}$ is implied by feasibility and then remaining elements are implied by sd-envy-freeness and feasibility.

Claim 3: Let $P^3 = (P_1, \hat{P}_2, P_3, \cdots, P_n)$. Then $\varphi(P^3)$ is as follows.

$$\begin{array}{rccccccc}
 & A_1 & A_2 & A_3 & A_4 & \cdots & A_K & \cdots \\
1: & \frac{d_1}{n-1} & \frac{d_2-\frac{d_1+d_2}{n}}{n-1} & \frac{d_3}{n} & \frac{d_4}{n} & \cdots & \frac{d_K}{n} & 0 \\
2: & 0 & \frac{d_1+d_2}{n} & \frac{d_3}{n} & \frac{d_4}{n} & \cdots & \frac{d_K}{n} & 0 \\
3\cdots n: & \frac{d_1}{n-1} & \frac{d_2-\frac{d_1+d_2}{n}}{n-1} & \frac{d_3}{n} & \frac{d_4}{n} & \cdots & \frac{d_K}{n} & 0
\end{array}$$

From P^1 to P^3, agent 2 is the unilateral deviator and she reversed the ranking of A_1 and A_2 with no other change. Hence sd-strategy-proofness implies $\varphi_{2A}(P^3) = \varphi_{2A}(P^1)$ for all $A \neq A_1, A_2$. Then sd-envy-freeness implies $\varphi_{iA}(P^3) = \varphi_{2A}(P^3)$ for all $i = 1, 3, \cdots, n$ and all $A \neq A_1, A_2$. In addition, sd-efficiency implies $\varphi_{2A_1}(P^3) = 0$, given which all other elements are implied by sd-envy-freeness and feasibility.

Claim 4: Let $P^4 = (\bar{P}_1, \hat{P}_2, P_3, \cdots, P_n)$. Then $\varphi(P^4)$ is as follows.

$$\begin{array}{rccccccc}
 & A_1 & A_2 & A_3 & A_4 & \cdots & A_K & \cdots \\
1: & \frac{d_1}{n-1} & 0 & \frac{d_1+d_2+d_3}{n} - \frac{d_1}{n-1} & \frac{d_4}{n} & \cdots & \frac{d_K}{n} & 0 \\
2: & 0 & \frac{d_1+d_2+d_3}{n} - \frac{d_3-\frac{d_2+d_3}{n}}{n-1} & \frac{d_3-\frac{d_2+d_3}{n}}{n-1} & \frac{d_4}{n} & \cdots & \frac{d_K}{n} & 0 \\
3\cdots n: & \frac{d_1}{n-1} & \frac{d_1+d_2+d_3}{n} - \frac{d_1}{n-1} - \frac{d_3-\frac{d_2+d_3}{n}}{n-1} & \frac{d_3-\frac{d_2+d_3}{n}}{n-1} & \frac{d_4}{n} & \cdots & \frac{d_K}{n} & 0
\end{array}$$

First, from P^3 to P^4, agent 1 is the unilateral deviator and she reversed the ranking of A_2 and A_3. Hence $\varphi_{1A}(P^4) = \varphi_{1A}(P^3)$ for all $A \neq A_2, A_3$. Second, from P^2 to P^4, agent 2 is the unilateral deviator and she reversed the ranking of A_1 and A_2. Hence $\varphi_{2A}(P^4) = \varphi_{2A}(P^2)$ for all $A \neq A_1, A_2$. Third, sd-envy-freeness implies that $\varphi_{iA}(P^4) = \varphi_{1A}(P^4)$ for all $i = 3, \cdots, n$ and $A \neq A_2, A_3$. In addition sd-envy-freeness implies also that $\varphi_{iA_3}(P^4) = \varphi_{2A_3}(P^4)$ for all $i = 3, \cdots, n$. Fourth, sd-efficiency implies $\varphi_{1A_2}(P^4) = \varphi_{2A_1}(P^4) = 0$. Last, the remaining elements, i.e., $\varphi_{1A_3}(P^4)$ and $\varphi_{iA_2}(P^4)$ for $i = 2, \cdots, n$, are implied by feasibility.

By the fact that there exists $x \in A_1 \cap A_2$ such that $x \notin A_k$ for all $k = 3, \cdots, K$ and that $q_x = d_1 + d_2$, we have the following contradiction.

$$d_1 + d_2 = \sum_{i \in I} \varphi_{iA_1}(P^4) + \varphi_{iA_2}(P^4) \Rightarrow d_1 = 0.$$

■

Remark 3. For the cases where $K = 2$ or 3, the answer is unclear and deserves further investigation. Let for example $K = 2$. In other words, all the objects have the same capacity. Let it be $\bar{q}$. Then whether or not there is a desirable rule on the universal domain depends on the parameters of the problem: m, $\bar{q}$, and n. When $n \leqslant 3$ or $m \leqslant 2$, the random serial dictatorship rule is sd-efficient and hence possibility holds. The arguments used to prove sd-efficiency of the RSD rule of objects with no more than 3 agents applies to show the result here. However, when $n \geqslant 4$ and $m \geqslant 3$, the conclusion depends on the relative size of $\bar{q}$ and n. Consider two extreme cases for example. If $\bar{q} = 1$, then impossibility holds. For this case, the idea used in the random assignment of objects can be applied to prove the impossibility. If $\bar{q} = n - 1$, the random serial dictatorship rule is sd-efficient. To see this, notice that the feasibility dictates that the serial dictatorship with an arbitrary ordering of agents will allocate to $n - 2$ agents the whole bundle and the remaining two agents share the other whole bundle. Then the problem degenerates to showing sd-efficiency with less than 3 agents, where the arguments for the object assignments apply. ■

5 Decomposable Random Assignments

We address in this section the decomposability issue, i.e., the problem of expressing a random assignment as a convex combination of deterministic assignments. We present a necessary condition, which is used to show that the PSB rule is in general not decomposable. In Appendix B, we transform the decomposability problem to n maximum flow problems, so that our knowledge on finding the maximum flow can be used to find the decompositions, if any.

For the random assignment of objects, a random assignment is a bi-stochastic matrix, i.e., a square matrix such that every element is in $[0, 1]$, every row sums to one and every column sums to one. The Birkhoff-von Neumann theorem states that every bi-stochastic matrix can be decomposed as a lottery over permutation matrices, which coincide with the set of deterministic assignments in that model. Hence decomposability is ensured in the classical framework. Unfortunately this is not true for the random assignment of bundles, and it is well-known that not every random assignment is decomposable. A random assignment studied here is also known in the literature as a plane-stochastic matrix. The issue of the decomposability has been studied by for example Brualdi and Csima (1975b), Brualdi and Csima (1975a), Csima (1970), Jurkat and Ryser (1968), Marchi and Tarazaga (1979).

Before presenting our study, we briefly discuss the papers studying the decomposability issue. Budish et al. (2013) studied two dimensional real matrices of size $|I| \times |O|$, where I denotes the agent set and O the object set, with the constraints in the form of $\underline{q}_S \leqslant \sum_{s \in S} L_s \leqslant \bar{q}_S$, where $S \subset I \times O$ is a subset of indexes and $\underline{q}_S, \bar{q}_S$ are two real numbers. These subsets of indexes are collected and called a constraint structure, denoted as $\mathcal{S}$. The authors then introduce a condition on $\mathcal{S}$ called "bihierarchy", which requires that $\mathcal{S}$ can be partitioned into two subsets $\mathcal{S}_1, \mathcal{S}_2$ so that, for an arbitrary pair S, S' from either $\mathcal{S}_1$ or $\mathcal{S}_2$, either they are disjoint ($S \cap S' = \emptyset$) or one contains the other ($S \subset S'$ or $S' \subset S$). They show that whenever the constraint structure is a bihierarchy, the random assignment is decomposable. Loosely speaking, bihierarchy requires that $\mathcal{S}$ can be partitioned into one set of "row constraints" and one set of "column constraints." A row(column) constraint refers to a subset of rows(columns) or a subset of a row(column). It is evident that the set of matrices attaining the bihierarchy structure includes the bi-stochastic matrices as a proper subset. It is also easy to see that a random assignment of bundles $L \in \mathcal{L}$ in our model may not attain the bihierarchy structure since two bundles might overlap with each other.

Given the impossibility of decomposability for general random assignments, some recent papers including Akbarpour and Nikzad (2017) and Nguyen et al. (2016) study extent to which a non-decomposable random assignment can be approximated by a decomposable one. Akbarpour and Nikzad (2017) push the result of Budish et al. (2013) further by showing that if the constraint structure $\mathcal{S}$ can be partitioned into two parts $\mathcal{P}$ and $\mathcal{Q}$ such that $\mathcal{P}$ is a bihierarchy and the elements in $\mathcal{Q}$ satisfy a condition defined with respect to $\mathcal{P}$, then there is a lottery over deterministic assignments such that the constraints in $\mathcal{P}$ are met exactly and the constraints in $\mathcal{Q}$ are met arbitrarily closely. This result is useful when some constraints are treated as "soft". For example, the constraint that a school has at least 50% of its students live within walking distance can be treated as soft, since if necessary 48% will also be satisfactory. Nguyen et al. (2016) impose a restriction on the random assignment that no agent gets a bundle containing more than k objects and show that it can be expressed as a lottery over deterministic assignments, which over-allocates each object by at most $k-1$ units.

We first provide a necessary condition for a random assignment to be decomposable. Given an arbitrary agent $i \in I$ and an arbitrary bundle $A \in \mathcal{X}$, we denote $\mathcal{D}_{iA}$ as the collection of deterministic assignments, each of which assigns A to agent i, i.e., $\mathcal{D}_{iA} \equiv \{D \in \mathcal{D} : D_{iA} = 1\}$. The following is the necessary condition.

Lemma 2. *Let $L \in \mathcal{L}$ be decomposable. For all $i \in I$ and $A \in \mathcal{X}$,*

$$L_{iA} \leqslant \sum_{D \in \mathcal{D}_{iA}} \min\{L_{jB} : D_{jB} = 1\}. \tag{2}$$

Proof. Let $L = \sum_{D \in \mathcal{D}} \beta(D) \cdot D$. Fix $i \in I$ and $A \in \mathcal{X}$, then $L_{iA} = \sum_{D \in \mathcal{D}_{iA}} \beta(D)$. Hence it suffices to show $\beta(D) \leqslant \min\{L_{jB} : D_{jB} = 1\}$ for all $D \in \mathcal{D}_{iA}$. Suppose not, let $\beta(D) > L_{jB}$ for some $j \neq i$ or $B \neq A$ and $D \in \mathcal{D}_{iA}$ such that $D_{jB} = 1$. Then we have a contradiction:

$$\beta(D) > L_{jB} = \sum_{D' \in \mathcal{D}_{jB}} \beta(D') \geqslant \beta(D)$$

where the last inequality comes from $D \in \mathcal{D}_{iA} \cap \mathcal{D}_{jB}$. ■

The following example uses the necessary condition to show that the PSB rule is in general not decomposable.

Example 15. *Let $X = \{a, b, c\}$, $q_x = 1\ \forall x \in X$, and $I = \{1, 2, 3\}$. Let the preference profile P and the random assignment $L = PSB(P)$ be as below.*

							ab	a	b	c	$\emptyset$
$P_1:$	ab	c	b	$\emptyset$	$\cdots$	$L_1:$	$3/4$	0	0	0	$1/4$
$P_2:$	c	b	a	$\emptyset$	$\cdots$	$L_2:$	0	0	$1/4$	$1/2$	$1/4$
$P_3:$	c	a	b	$\emptyset$	$\cdots$	$L_3:$	0	$1/4$	0	$1/2$	$1/4$

Utilizing Lemma 2, we can easily see that L is not decomposable. Let D^1 be such that $D^1_{1ab} = D^1_{2c} = D^1_{3\emptyset} = 1$. Let D^2 be such that $D^2_{1ab} = D^2_{2\emptyset} = D^2_{3c} = 1$. Then $\mathcal{D}_{1ab} = \{D^1, D^2\}$

since if bundle ab is given to agent 1, either agent 2 or 3 gets c and the other gets nothing. The following simple calculation tells that L is not decomposable.

$$L_{1ab} = 3/4 > \sum_{D \in \mathcal{D}_{1ab}} \min \{L_{jB} : D_{jB} = 1\}$$
$$= \min\{L_{1ab}, L_{2c}, L_{3\emptyset}\} + \min\{L_{1ab}, L_{2\emptyset}, L_{3c}\} = 1/4 + 1/4.$$

■

In Appendix B, we transform a decomposition problem into a series of maximum flow problems. Since the maximum flow problem has long been studied by mathematicians and computer scientists, such a transformation is expected to be useful in the sense that already known results on maximum flows can be applied to check decomposability. In particular, when we need to judge whether or not a given random assignment is decomposable, one generated by the PSB rule for instance, some program packages can be used immediately. For example, MatLab provides built-in functions to calculate maximum flows.

6 Conclusion

We study random assignments of bundles with no free disposal. The induced feasibility requirement has been shown to have significant implications for the design of random assignment rules. First, the characterization of sd-efficiency is fundamentally different in this setup. Second, the possibility result of Liu (2017) fails under this new feasibility requirement. However, we identify a preference domain on which a desirable existence result exists.

The following questions remain unresolved and are presumably of interest. First, given an arbitrary preference profile, what is an algorithm that identifies all sd-efficient assignments? Second, on the essentially monotonic domain, is there a desirable, in particular a non-constant rule, that is different from the RSDB rule (equivalently PSB rule)?

Appendix

Appendix A presents the the proof of Theorem 1 and the definition of the simultaneous eating algorithm for bundles. Thereafter, Appendix B transforms the decomposability problem into a series of maximum flow problems.

A Efficiency

A.1 Proof of Theorem 1

Necessity: We show the contrapositive statement. Let $L \in \mathcal{L}$ be balanced at $P \in \mathbb{P}^n$. Then there is an $\alpha : \mathcal{T} \to \mathbb{R}_+$ such that (i) $\alpha(i, A, B) > 0$ implies $L_{iA} > 0$ and $B \; P_i \; A$, (ii) $\forall x \in X$: $\sum_{\{(i,A,B) \in \mathcal{T} : x \in B\}} \alpha(i, A, B) = \sum_{\{(i,A,B) \in \mathcal{T} : x \in A\}} \alpha(i, A, B)$. We show L is not sd-efficient at P.

To do this, we construct another matrix L'. Let $\epsilon \in \mathbb{R}_{++}$ be a very small positive number and let for $\forall j \in I$ and $\forall C \in \mathcal{X}$,

$$L'_{jC} = L_{jC} + \sum_{\{(i,A,B)\in\mathcal{T}:i=j,B=C\}} \epsilon \cdot \alpha(i,A,B) - \sum_{\{(i,A,B)\in\mathcal{T}:i=j,A=C\}} \epsilon \cdot \alpha(i,A,B).$$

Notice first that by letting ϵ to be sufficiently small, $L'_{jC} \geqslant 0$ for all $j \in I$ and $C \in \mathcal{X}$. Then the following two classes of equations show that L' is a feasible random assignment, in other words $L' \in \mathcal{L}$.

$$\begin{aligned}
\forall j \in I: \sum_{C\in\mathcal{X}} L'_{jC} &= \sum_{C\in\mathcal{X}} L_{jC} + \sum_{C\in\mathcal{X}} \sum_{\{(i,A,B)\in\mathcal{T}:i=j,B=C\}} \epsilon \cdot \alpha(i,A,B) \\
&\quad - \sum_{C\in\mathcal{X}} \sum_{\{(i,A,B)\in\mathcal{T}:i=j,A=C\}} \epsilon \cdot \alpha(i,A,B) \\
&= 1 + \sum_{\{(i,A,B)\in\mathcal{T}(P,L):i=j\}} \epsilon \cdot [\alpha(i,A,B) - \alpha(i,A,B)] = 1 \\
\forall x \in X: \sum_{i\in I, x\in C} L'_{jC} &= \sum_{i\in I, x\in C} L_{jC} + \sum_{\{(i,A,B)\in\mathcal{T}:x\in B\}} \epsilon \cdot \alpha(i,A,B) \\
&\quad - \sum_{\{(i,A,B)\in\mathcal{T}:x\in A\}} \epsilon \cdot \alpha(i,A,B) \\
&= \sum_{i\in I, x\in C} L_{jC} = q_x.
\end{aligned}$$

The last equality follows from the definition of α. Next, it is evident that $L'_j \; P^{sd}_j \; L_j$ for all $j \in I$ since the probability transfers are all from less preferred bundles to more preferred ones. Consequently, L is dominated by L' constructed above and hence not sd-efficient at P.

Sufficiency: We show the contrapositive statement. Let L be sd-inefficient at P. Then there is another random assignment $L' \in \mathcal{L}$ such that $L' \neq L$ and $L'_j \; P^{sd}_j \; L_j$ for all j. We construct an $\alpha : \mathcal{T}(P,L) \to \mathbb{R}_+$ such that (i) $\alpha(i,A,B) > 0$ implies $L_{iA} > 0$ and $B \; P_i \; A$, (ii) $\forall x \in X$: $\sum_{\{(i,A,B)\in\mathcal{T}:x\in B\}} \alpha(i,A,B) = \sum_{\{(i,A,B)\in\mathcal{T}:x\in A\}} \alpha(i,A,B)$.

By the fact that $L_j, L'_j \in \Delta(\mathcal{X})$ for all $j \in I$, there is a system of probability transfers $\beta : \mathcal{T} \to \mathbb{R}_+$ such that, $\forall j \in I$ and $C \in \mathcal{X}$,

$$L'_{jC} = L_{jC} + \sum_{\{(i,A,B)\in\mathcal{T}:i=j,B=C\}} \beta(i,A,B) - \sum_{\{(i,A,B)\in\mathcal{T}:i=j,A=C\}} \beta(i,A,B). \tag{3}$$

In other words, L' is constructed from L by β. In addition, since both L and L' are feasible random assignments, $\forall x \in X$,

$$\sum_{\{(i,A,B)\in\mathcal{T}:x\in B\}} \beta(i,A,B) = \sum_{\{(i,A,B)\in\mathcal{T}:x\in A\}} \beta(i,A,B). \tag{4}$$

Notice that the vector β in general can not serve as the α we want since β may transfer some positive probability from (i,A) to (i,B) where $L_{iA} = 0$ and(or) $A \; P_i \; B$, which is not allowed by the definition of α. In the following, we construct a wanted α from β in two steps.

Step 1: Given $\beta : \mathcal{T} \to \mathbb{R}_+$ satisfying (3) and (4), we construct a $\gamma : \mathcal{T} \to \mathbb{R}_+$ satisfying not only (3) and (4) but also that $\gamma(i,A,B) > 0$ implies $L_{iA} > 0$.

To do so, pick an arbitrary $(i,A,B) \in \mathcal{T}$ such that $\beta(i,A,B) > 0$ and $L_{iA} = 0$. We claim that there is another bundle $C \neq A$ such that $\beta(i,C,A) > 0$, since otherwise, according to (3), we have a contradiction: $L'_{iA} \leqslant 0 + 0 - \beta(i,A,B) < 0$.

Let $\beta(i, A, B) = u$ and $\beta(i, C, A) = v$. Then we update β to β' by the following three changes without anything else.

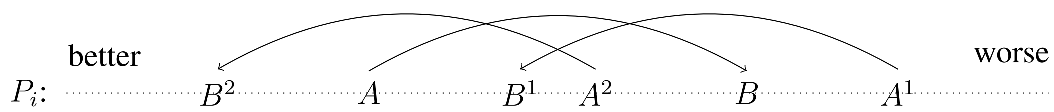
$$\text{If } u \leqslant v, \text{ let } \beta'(i, A, B) = 0, \beta'(i, C, B) = \beta(i, C, B) + u, \beta'(i, C, A) = v - u;$$

$$\text{If } u > v, \text{ let } \beta'(i, A, B) = u - v, \beta'(i, C, B) = \beta(i, C, B) + v, \beta'(i, C, A) = 0.$$

Notice first that no matter whether $u \leqslant v$ or not, β' still satisfies (3) and (4). Notice in addition that for the case where $u \leqslant v$, $\beta'(i, A, B) = 0$ and hence the unwanted instance where $\beta(i, A, B) > 0$ and $L_{iA} = 0$ is eliminated. While for the other case where $u > v$, this unwanted instance is still there. Then we can repeat the update above by finding some other bundle $D \neq A$ such that $\beta'(i, D, A) > 0$. By repeatedly applying the above update, we can finally construct a vector $\gamma : \mathcal{T} \to \mathbb{R}_+$ which satisfies not only (3) and (4) but also that $\gamma(i, A, B) > 0$ implies $L_{iA} > 0$.

Step 2: Given $\gamma : \mathcal{T} \to \mathbb{R}_+$ generated by the last step, we construct the wanted $\alpha : \mathcal{T} \to \mathbb{R}_+$ satisfying not only (3) and (4) but also that $\alpha(i, A, B) > 0$ implies $L_{iA} > 0$ and $B\ P_i\ A$. In other words, positive probability transfers are allowed only from less preferred bundles to more preferred bundles.

To do so, pick an arbitrary $(i, A, B) \in \mathcal{T}$ such that $\gamma(i, A, B) > 0$ and $A\ P_i\ B$. We claim that there is a sequence $(i, A^l, B^l)_{l=1}^{L} \subset \mathcal{T}$ such that (i) $\gamma(i, A^l, B^l) > 0$ and $B^l\ P_i\ A^l$ for all $l = 1, \cdots, L$, (ii) $B^1\ R_i\ B\ R_i\ A^1$, (iii) $B^l\ R_i\ A^{l+1}$ for all $l = 1, \cdots, L-1$, and (iv) $B^L\ R_i\ A\ R_i\ A^L$. The following figure depicts an instance where $L = 2$.

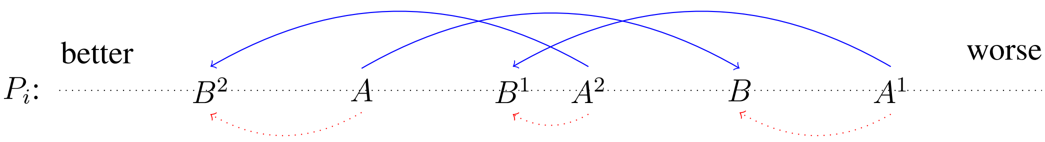


We show the existence of such a sequence by construction. First, notice that there exists $(i, A^1, B^1) \in \mathcal{T}$ such that $\gamma(i, A^1, B^1) > 0$, $B^1\ P_i\ A^1$, and $B^1\ R_i\ B\ R_i\ A^1$. Because otherwise, $\sum_{C P_i B} L'_{iC} < \sum_{C P_i B} L_{iC}$ which is contradicting $L'_i\ P_i^{sd}\ L_i$. Fixing A^1 and B^1, if $B^1\ R_i\ A$, we are done by letting $L = 1$. If not, there exists another triple $(i, A^2, B^2) \in \mathcal{T}$ such that $\gamma(i, A^2, B^2) > 0$, $B^2\ P_i\ A^2$, and $B^1\ R_i\ A^2$. Because otherwise, $\sum_{C P_i B^1} L'_{iC} < \sum_{C P_i B^1} L_{iC}$ which is contradicting $L'_i\ P_i^{sd}\ L_i$. We repeat this procedure to identify the sequence and finally the finiteness of bundles gives $B^L\ R_i\ A\ R_i\ A^L$.

Fixing such a sequence, let $\mu = \min\{\gamma(i, A, B), \gamma(i, A^l, B^l) : l = 1, \cdots, L\}$. Then we update γ to γ' by the following changes without anything else.

$$\begin{aligned}
\gamma'(i, A, B) &= \gamma(i, A, B) - \mu \\
\gamma'(i, A^l, B^l) &= \gamma(i, A^l, B^l) - \mu, \quad \forall l = 1, \cdots, L \\
\gamma'(i, A, B^L) &= \gamma(i, A, B^L) + \mu \\
\gamma'(i, A^1, B) &= \gamma(i, A^1, B) + \mu \\
\gamma'(i, A^{l+1}, B^l) &= \gamma(i, A^{l+1}, B^l) + \mu, \quad \forall l = 1, \cdots, L-1
\end{aligned}$$

The following figure depicts the update, where blue solid arrows corresponds to the minuses and the red dotted arrows to the pluses.



It is evident that $\gamma : \mathcal{T}(L) \to \mathbb{R}_+$ satisfies not only (3) and (4) but also that $\gamma'(i, A, B) > 0$ implies $L_{iA} > 0$. It satisfies (3) evidently. To see that it satisfies (4), notice that for all $C \in \mathcal{X}$, $\sum_{\{(i,A,B)\in\mathcal{T}:A=C\}} \gamma'(i, A, B) = \sum_{\{(i,A,B)\in\mathcal{T}:A=C\}} \gamma(i, A, B)$ and $\sum_{\{(i,A,B)\in\mathcal{T}:B=C\}} \gamma'(i, A, B) = \sum_{\{(i,A,B)\in\mathcal{T}:B=C\}} \gamma(i, A, B)$.

By repeatedly applying the above update, we can finally construct an $\alpha : \mathcal{T} \to \mathbb{R}_+$ satisfying not only (3) and (4) but also that $\alpha(i, A, B) > 0$ implies $L_{iA} > 0$ and $B\ P_i\ A$.

To show the sufficiency, it remains to show that such an α satisfies$\forall x \in X$:

$$\sum_{\{(i,A,B)\in\mathcal{T}:x\in B\}} \alpha(i, A, B) = \sum_{\{(i,A,B)\in\mathcal{T}:x\in A\}} \alpha(i, A, B),$$

which is evident by the fact that α satisfies (4).

A.2 Simultaneous Eating Algorithm with Varying Speeds

This section is devoted to the relation between sd-efficiency and the simultaneous eating algorithm with varying eating speeds, which are variants of the PSB rule by allowing non-uniform eating speeds. Let $w_i : [0, 1] \to \mathbb{R}_+$ with $\int_0^1 w_i(t)dt = 1$ denote the eating speed for agent i, where $w_i(t)$ denote the eating speed of agent i at time t. The restriction that the integral of $w_i(t)$ on the interval $[0, 1]$ equals one comes from the fact that the summation of the shares eaten by an agent should sum to one. PSB is a special case of the simultaneous eating with $w_i(t) = 1$ for all $i \in I$ and $t \in [0, 1]$. Given a speed profile $w = (w_i)_{i\in I}$, we define the simultaneous eating algorithm for bundles below.

Definition 14. *Given a preference profile $P \in \mathbb{P}^n$, the **simultaneous eating algorithm for bundles with speed profile** w **(SEB^w)** gives a random assignment $SEB^w(P) \equiv L^{\bar{v}}$ below.*

Let $t^0 = 0$, $\mathcal{X}^0 = \mathcal{X}$, $r^0_x = q_x$ and $r^0_{\bar{x}} = n - q_x$ for all $x \in X$. Let in addition L^0 be a matrix of size $n \times |\mathcal{X}|$ with all zeros.

For $v = 1, \cdots, \bar{v}$

$$I^v_x \equiv \{i \in I : x \in \tau(P_i, \mathcal{X}^{v-1})\}, \forall x \in X;$$

$$t^v \equiv t^{v-1} + \min \left\{ \begin{array}{l} \left\{\delta \in \mathbb{R}_+ : \sum_{i\in I^v_x} \int_{t^{v-1}}^{t^{v-1}+\delta} w_i(t)dt = r^{v-1}_x, r^{v-1}_x > 0\right\} \\ \bigcup \left\{\delta \in \mathbb{R}_+ : \sum_{i\in I\setminus I^v_x} \int_{t^{v-1}}^{t^{v-1}+\delta} w_i(t)dt = r^{v-1}_{\bar{x}}, r^{v-1}_{\bar{x}} > 0\right\} \end{array} \right\};$$

$$L^v_{iA} \equiv L^{v-1}_{iA} + \begin{cases} \int_{t^{v-1}}^{t^v} w_i(t)dt, & \textit{if } A = \tau(P_i, \mathcal{X}^{v-1}) \\ 0, & \textit{otherwise} \end{cases}, \forall i \in I, A \in \mathcal{X};$$

$$r_x^v \equiv r_x^{v-1} - \sum_{i \in I_x^v} \int_{t^{v-1}}^{t^v} w_i(t)dt, \forall x \in X;$$

$$r_{\bar{x}}^v \equiv r_{\bar{x}}^{v-1} - \sum_{i \in I \setminus I_x^v} \int_{t^{v-1}}^{t^v} w_i(t)dt, \forall x \in X;$$

$$\mathcal{X}^v \equiv \mathcal{X}^{v-1} \setminus \{A \in \mathcal{X}^{v-1} : \exists x \in X \textit{ s.t. } [x \in A, r_x^v = 0] \textit{ or } [x \notin A, r_{\bar{x}}^v = 0]\};$$

where $\bar{v}$ is identified by the step when $\mathcal{X}^{\bar{v}} = \emptyset$.

Proposition 5. *The random assignment $L = SEB^w(P)$ is sd-efficient at P for all w.*

The above proposition can be proved by the same argument that proves sd-efficiency for the PSB rule.

Remark 4. The converse of Proposition 5 is not true. In other words, there is an sd-efficient L at a given preference profile P that can not be generated by the simultaneous eating algorithm with any eating speed. Let $X = \{a, b, c\}$ and $q_x = 1$ for all $x \in X$. Let the preference profile P and the assignment L be as below.

$$\begin{array}{lccc} P_1: & ab & c & \cdots \\ P_2: & c & b & \cdots \\ P_3: & c & a & \cdots \end{array} \qquad \begin{array}{lcccc} & a & b & c & ab \\ L_1: & 0 & 0 & 1 & 0 \\ L_2: & 0 & 1 & 0 & 0 \\ L_3: & 1 & 0 & 0 & 0 \end{array}$$

The random assignment L has been verified to be sd-efficient at P by the Step 1.2 in the proof of Proposition 1. We now show that there is no speed profile w such that $L = SEB^w(P)$. Suppose not, let for each $x \in X$, $t(x)$ be the point in time when object x is eaten up. Then $L_{1ab} = 0$ implies that at the earliest time when agent 1 has a positive eating speed, i.e., $\min\{t \in (0, 1] : w_1(t) > 0\}$, either a or b has already been eaten up, i.e., $\min\{t \in (0, 1] : w_1(t) > 0\} > \min\{t(a), t(b)\}$. Notice that $L_{1c} > 0$, hence $t(c) > \min\{t \in [0, 1] : w_1(t) > 0\} > \min\{t(a), t(b)\}$. If $\min\{t(a), t(b)\} = t(b)$, then $t(c) > t(b)$ implies $L_{2b} = 0$: contradiction. Otherwise, $t(c) > t(a)$ implies $L_{3a} = 0$: contradiction. ■

B Decomposability and Maximum Flow Problems

Let $L \in \mathcal{L}$ be an arbitrarily given random assignment and let $\beta \in \Delta(\mathcal{D})$ be an arbitrary lottery over deterministic assignments. We show in the following that judging whether β is a decomposition of L is equivalent to checking n maximum flow problems, one for each $i \in I$. The primitive of such a problem is a single-source multiple-sink flow network, denoted as $N_i(L, \beta) \equiv (V, E, c)$.

- The set of vertices V is the union of three sets: $V = I \cup (I \times \mathcal{X}) \cup \mathcal{D}$.
- The set of edges $E = \{(i, jA) : i = j \text{ and } A \in \mathcal{X}\} \cup \{(iA, D) : A \in \mathcal{X} \text{ and } D_{iA} = 1\} \cup \{(D, jB) : j \neq i, B \in \mathcal{X}, \text{ and } D_{jB} = 1\} \cup \{(jB, j) : j \neq i \text{ and } B \in \mathcal{X}\}$. Hence a

typical flow starts from i to a vertex iA, then to a vertex D such that $D_{iA} = 1$, then to a jB such that $D_{jB} = 1$, and finally to j. In other words, the unique source is i and the set of sinks is $I \backslash \{i\}$.

- The capacity function $c : (I \times \mathcal{X}) \cup \mathcal{D} \to \mathbb{R}_+$ specifies for each vertex other than the source and the sinks a capacity of flow. In particular, $c(iA) = L_{iA}$ for all $iA \in I \times \mathcal{X}$ and $c(D) = \beta(D)$ for all $D \in \mathcal{D}$. So the parameters of this maximum problem, β and L, specify the capacities of vertices.

Below is an example of such a network.

Example 16. *Consider the random assignment L in Example 3. Figure 8 depicts $N_1(L, \beta)$ for an arbitrarily specified β.* ■

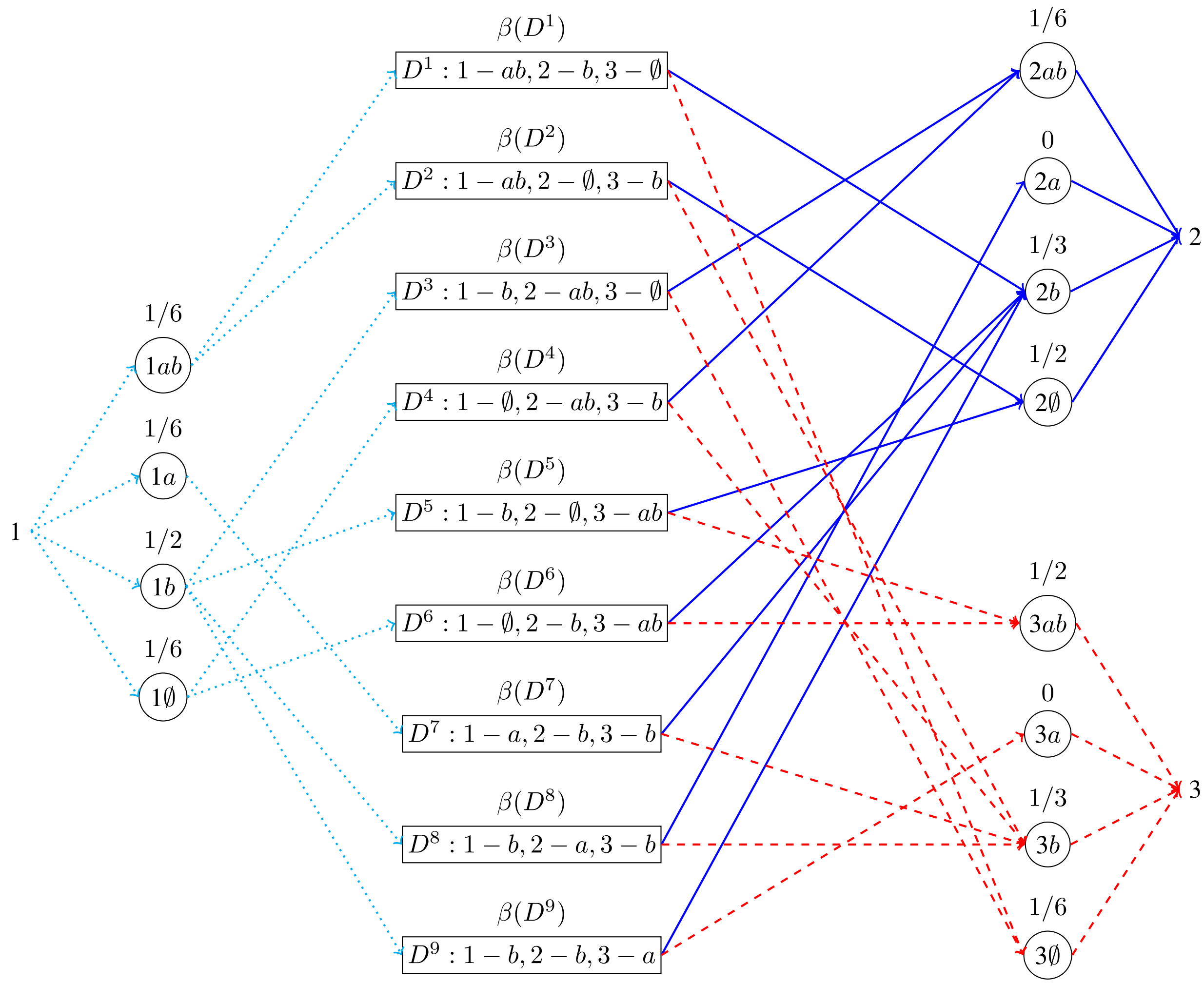


Figure 8: A Maximum Flow Problem

It is evident that the lottery for agent i induced by β equals L_i if and only if the maximum flow on the network $N_i(L, \beta)$ is one. This observation gives the following lemma.

Lemma 3. *A lottery $\beta \in \mathcal{D}$ is a decomposition of $L \in \mathcal{L}$ if and only if the maximum flow on $N_i(L, \beta)$ is 1 for all $i \in I$.*

The following is an application of the above lemma.

Example 16 (continued)**.** *Let $\beta \in \Delta(\mathcal{D})$ denote an arbitrary lottery on deterministic assignments. From the network in Example 16, it is seen that $maxflow N_i(L, \beta) = 1$ is equivalent to the equation system i as below for each $i = 1, 2, 3$.*

system 1:	*system 2:*	*system 3:*
$\beta(D^1) + \beta(D^2) = 1/6$	$\beta(D^3) + \beta(D^4) = 1/6$	$\beta(D^5) + \beta(D^6) = 1/2$
$\beta(D^7) = 1/6$	$\beta(D^8) = 0$	$\beta(D^9) = 0$
$\beta(D^3) + \beta(D^5) + \beta(D^8) + \beta(D^9) = 1/2$	$\beta(D^1) + \beta(D^6) + \beta(D^7) + \beta(D^9) = 1/3$	$\beta(D^2) + \beta(D^4) + \beta(D^7) + \beta(D^8) = 1/3$
$\beta(D^4) + \beta(D^6) = 1/6$	$\beta(D^2) + \beta(D^5) = 1/2$	$\beta(D^1) + \beta(D^3) = 1/6$

By solving the above equation system, we get the unique decomposition of the random assignment: $\beta(D^1) = 1/6$, $\beta(D^4) = 1/6$, $\beta(D^5) = 1/2$, $\beta(D^7) = 1/6$, and $\beta(D^2) = \beta(D^3) = \beta(D^6) = \beta(D^8) = \beta(D^9) = 0$. ■

References

ABDULKADIROĞLU, A. AND T. SÖNMEZ (1998): "Random serial dictatorship and the core from random endowments in house allocation problems," *Econometrica*, 689–701.

AKBARPOUR, M. AND A. NIKZAD (2017): "Approximate random allocation mechanisms," .

BOGOMOLNAIA, A. AND H. MOULIN (2001): "A new solution to the random assignment problem," *Journal of Economic Theory*, 100, 295–328.

BRUALDI, R. AND J. CSIMA (1975a): "Extremal plane stochastic matrices of dimension three," *Linear Algebra and its Applications*, 11, 105–133.

BRUALDI, R. A. AND J. CSIMA (1975b): "Stochastic patterns," *Journal of Combinatorial Theory, Series A*, 19, 1–12.

BUDISH, E. (2011): "The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes," *Journal of Political Economy*, 119, 1061–1103.

BUDISH, E. AND E. CANTILLON (2012): "The multi-unit assignment problem: Theory and evidence from course allocation at Harvard," *American Economic Review*, 102, 2237–71.

BUDISH, E., Y.-K. CHE, F. KOJIMA, AND P. MILGROM (2013): "Designing random allocation mechanisms: Theory and applications," *The American Economic Review*, 103, 585–623.

CHANG, H.-I. AND Y. CHUN (2017): "Probabilistic assignment of indivisible objects when agents have the same preferences except the ordinal ranking of one object," *Mathematical Social Sciences*, 90, 80–92.

CSIMA, J. (1970): "Multidimensional stochastic matrices and patterns," *Journal of Algebra*, 14, 194–202.

HYLLAND, A. AND R. ZECKHAUSER (1979): "The efficient allocation of individuals to positions," *Journal of Political economy*, 87, 293–314.

JURKAT, W. AND H. RYSER (1968): "Extremal configurations and decomposition theorems. I," *Journal of Algebra*, 8, 194–222.

KASAJIMA, Y. (2013): "Probabilistic assignment of indivisible goods with single-peaked preferences," *Social Choice and Welfare*, 41, 203–215.

LE BRETON, M. AND A. SEN (1999): "Separable preferences, strategyproofness, and decomposability," *Econometrica*, 67, 605–628.

LIU, P. (2017): "Random assignments on sequentially dichotomous domains," *Working Paper*.

LIU, P. AND H. ZENG (2017): "Random assignments on preference domains with a tier structure," *SMU Working Paper*.

MARCHI, E. AND P. TARAZAGA (1979): "About (k, n) stochastic matrices," *Linear Algebra and its Applications*, 26, 15–30.

NGUYEN, T., A. PEIVANDI, AND R. VOHRA (2016): "Assignment problems with complementarities," *Journal of Economic Theory*, 165, 209–241.

PÁPAI, S. (2000a): "Strategyproof assignment by hierarchical exchange," *Econometrica*, 68, 1403–1433.

——— (2000b): "Strategyproof multiple assignment using quotas," *Review of Economic Design*, 5, 91–105.

——— (2001): "Strategyproof and nonbossy multiple assignments," *Journal of Public Economic Theory*, 3, 257–271.

SÖNMEZ, T. AND M. U. ÜNVER (2010): "Course bidding at business schools," *International Economic Review*, 51, 99–123.

SVENSSON, L.-G. (1999): "Strategy-proof allocation of indivisible goods," *Social Choice and Welfare*, 16, 557–567.