Random assignment with multi-unit demands

Haris Aziz

NICTA and UNSW, Kensington 2033, Australia

Abstract

We consider the setting in which agents express preferences over objects, objects are allocated to agents based on the preferences, and there may be more objects than agents. In randomized settings, agents need to reason about their random allocations. The most well-established preference relation to compare random allocations of objects is stochastic dominance (SD). We present two impossibility results that show that there exist no rules that are anonymous, neutral and weakly strategyproof or weakly group-strategyproof with respect to SD. For the random assignment problem in which each agent is allocated at most one object, the probabilistic serial mechanism (PS) was shown by Bogomolnaia and Moulin [8] to be SD envy-free, weak SD-strategyproof, and SD-efficient. If agents get multiple units of objects, Kojima [21] showed that (one-at-a-time) PS does not remain weak SD-strategyproof. In this paper, we study another generalization of PS for multiple units called multi-unit-eating PS which was defined by Che and Kojima [13]. We prove that multi-unit-eating PS satisfies SD-envy-freeness, weak SD-strategyproofness, and unanimity. Therefore, if maintaining the strategic and fairness properties is a requirement, then multi-unit-eating PS is the appropriate generalization of PS.

Keywords: Random assignment, fairness, efficiency, strategyproofness

JEL: C62, C63, and C78

1. Introduction

In the assignment problem, agents express linear preferences over objects and an object is assigned to each agent keeping in view the agents’ preferences. The problem models one of the most fundamental setting in computer science and economics with numerous applications [see e.g., 13, 24, 26, 22, 23, 8, 3]. Depending on the application setting, the objects could be car-park spaces, dormitory rooms, replacement kidneys, school seats, etc. The assignment problem is also referred to as house allocation [3, 2]. Since randomization is one of the oldest tools to achieve fairness, we consider the random assignment problem [see e.g., 18, 26, 8, 20, 16, 7, 11] in which an object is allocated randomly to each agent. In random settings, the most established preference relation between random assignments is stochastic dominance (SD). SD requires that one
random allocation is preferred to another one iff the former first-order stochastically dominates the latter. This relation is especially important because one random allocation stochastically dominates another one iff the former yields at least as much expected utility as the latter for any von-Neumann-Morgenstern (vNM) utility representation consistent with the ordinal preferences [see, e.g., 6]. The $SD$ relation can be used to define corresponding notions of envy-free, efficiency, and strategyproofness [8, 20]. In this paper, we check which levels of fairness, efficiency, and strategyproofness can be satisfied simultaneously.

For the random assignment problem without multi-unit demands, the most common and well-known way to assign objects is random priority ($RP$) in which a permutation of agents is chosen uniformly at random and agents successively take their most preferred available object [see e.g., 1, 8, 14]. Although $RP$ is strategyproof and results in a Pareto optimal assignment, Bogomolnaia and Moulin [8] in a remarkable paper showed that $RP$ does not satisfy the stronger efficiency notion of stochastic dominance ($SD$) efficiency and also a fairness concept called $SD$-envy-freeness. Furthermore, they presented an elegant algorithm called $PS$ (probabilistic serial) that is not only $SD$-efficient and $SD$-envy-free but also satisfies weak $SD$-strategyproofness. In $PS$, agents ‘eat’ the most favoured available object at the same rate until all the objects are consumed. The fraction of object consumed by an agent is the probability of the agent getting that object.

Since the work of Bogomolnaia and Moulin [8], $PS$ has received considerable attention and has been extended in a number of ways [see e.g., 20, 4, 25]. In particular, it can be naturally extended to the more general case with multi-unit demands in which there are $nc$ objects and $c \geq 1$ objects are allocated to each of the agents [see e.g., 8, 17, 21]. The extension does not require any modification to the specification of $PS$: agents continue eating their most preferred available object until all the objects have been consumed. Although this one-at-a-time extension (which we will refer to as $OPS$) still satisfies $SD$-efficiency and $SD$-envy-freeness, it is not weak $SD$-strategyproof [21]. Incidentally there is another extension of $PS$ called the multi-unit-eating probabilistic serial that was briefly described by Che and Kojima [13] but has received no attention in the literature. In multi-unit-eating $PS$, each agent tries to eat his $c$ most preferred objects that are still available at a uniform speed until all objects have been consumed. We show that multi-unit-eating $PS$ satisfies desirable properties and is the only known mechanism that is weak $SD$-strategyproof, $SD$-envy-free, and unanimous. We point out that the problem of discrete assignment with multi-unit demands has attracted considerable attention [see e.g., 10, 19, 9]. In this paper, we focus on random assignments with multi-unit demands.

Apart from $RP$ and $PS$, two other natural assignment rules are uniform...
assignment and priority. In the uniform assignment, each agent gets \(1/n\) of each object \([12, 21]\). In the priority mechanism, there is a permutation \(\pi\) of agents, and each agent in the permutation is assigned the \(c\) most desirable available objects. The priority mechanism is also referred to as serial dictator in the literature \([22, 23]\). Whereas uniform assignment does not take into account the preferences of agents and is highly inefficient, priority is highly unfair to the agents at the end of the permutation.

Contributions. We first prove that for multi-unit demands, there exists no anonymous, neutral, weak SD-strategyproof and SD-efficient random assignment rule. The statement is somewhat surprising considering that all the four axioms used in the statement are minimal requirements. Incidentally, we have not used SD envy-freeness that is often used to obtain characterizations or impossibility statements in the literature [see e.g., \([17, 8, 21]\) and is a very demanding requirement. The result is then extended to random assignment without multi-unit demands if requiring weak SD weak group-strategyproofness instead of weak SD strategyproofness.

We then conduct an axiomatic analysis of the multi-unit-eating PS. It is first highlighted that the definition of multi-unit-eating PS in the literature is not entirely correct. A proper definition of multi-unit-eating PS is formulated. We show that for multi-unit demands, in contrast to OPS, multi-unit-eating PS satisfies weak SD-strategyproofness. We prove that multi-unit-eating PS satisfies SD envy-freeness which is one of the strongest notions of fairness. On the other hand, multi-unit-eating PS does not fare well in terms of efficiency. We prove that multi-unit-eating PS does not even satisfy ex post efficiency although it does satisfy unanimity. Therefore when we generalize PS for multi-unit demands, OPS is the right extension if the focus is on efficiency. On the other hand multi-unit-eating PS is the right extension, if the aim is to maintain weak SD-strategyproofness. The arguments for weak SD-strategyproofness and SD envy-freeness of MPS multi-unit-eating PS also simplify the proofs for PS for single-unit demands in \([\text{3}]\). The study helps clarify the relative merits of different assignment rules for multi-unit demands. The relative merits of prominent random assignment rules are then summarized in Table 1 in the final section.

2. Preliminaries

Random assignment problem. A random assignment problem is a triple \((N, O, \succapprox)\) where \(N\) is the set of \(n\) agents \(\{1, \ldots, n\}\), \(O = \{o_1, \ldots, o_m\}\) is the set of objects, and \(\succapprox = (\succapprox_1, \ldots, \succapprox_n)\) specifies strict, complete, and transitive preferences \(\succapprox_i\) of agent \(i\) over \(O\). We will assume that \(m\) is a multiple of \(n\) i.e., \(m = nc\) where \(c\) is an integer. We will denote by \(\mathcal{R}(O)\) as the set of all complete and transitive relations over the set of objects \(O\).

A random assignment \(p\) is a \((n \times m)\) matrix \([p(i)(o_j)]\) such that for all \(i \in N\), and \(o_j \in O\), \(p(i)(o_j) \in [0, 1]\); \(\sum_{i \in N} p(i)(o_j) = 1\) for all \(j \in \{1, \ldots, n\}\); and \(\sum_{o \in O} p(i)(o_j) = c\) for all \(i \in N\). The value \(p(i)(o_j)\) represents the probability
of object $o_j$ being allocated to agent $i$. Each row $p(i) = (p(i)(o_1), \ldots, p(i)(o_m))$ represents the allocation of agent $i$. The set of columns correspond to the objects $o_1, \ldots, o_m$. A feasible random assignment is discrete if $p(i)(o) \in \{0,1\}$ for all $i \in N$ and $o \in O$. A random assignment rule specifies for each preferences profile a random assignment.

**Example 1.** Consider random assignment for two agents $N = \{1,2\}$ and four agents $O = \{o_1, o_2, o_3, o_4\}$. Let us that agent 1 gets $o_1$ with probability one, and objects $o_3$ and $o_4$ with probability half. Then the random assignment can be represented by the following matrix.

$$p = \begin{pmatrix} 1 & 0 & 1/2 & 1/2 \\ 0 & 1 & 1/2 & 1/2 \end{pmatrix}.$$ 

Given two random assignments $p$ and $q$, $p(i) \succ^SD_i q(i)$ i.e., a player $i$ $SD$ prefers allocation $p(i)$ to allocation $q(i)$ if $\sum_{o_j \in \{o_k \mid k \neq i,o\}} p(i)(o_j) \geq \sum_{o_j \in \{o_k \mid k \neq i,o\}} q(i)(o_j)$ for all $o \in O$.

Let $p(i)$ and $q(i)$ be two random allocations. Let $o \in O$ be the most preferred object such that $p(i)(o) \neq q(i)(o)$. Then, $p(i) \succ^DL_i q(i) \iff p(i)(o) > q(i)(o)$.

**Example 2.** Let us assume that agent one has the following preferences in Example [1] $o_1 \succ_1 o_2 \succ_1 o_3 \succ_1 o_4$ which can also be represented as $1 : o_1, o_2, o_3, o_4$. Then, $p(1) \succ^SD_1 p(2)$ and also $p(1) \succ^DL_1 p(2)$.

**Envy-freeness.** An assignment $p$ satisfies $SD$ envy-freeness if each agent (weakly) $SD$ prefers his allocation to that of any other agent: $p(i) \succ^SD_j p(j)$ for all $i, j \in N$. An assignment $p$ satisfies weak $SD$ envy-freeness if no agent strictly $SD$ prefers someone else’s allocation to his: $\neg[p(j) \succ^SD_i p(i)]$ for all $i, j \in N$. For fairness concepts, $SD$ envy-freeness implies weak $SD$-envy-freeness [8].

**Economic efficiency.** An assignment is perfect if each agents gets his most preferred $c$ objects. A discrete assignment $p$ is Pareto optimal if there does not exist another discrete assignment $q$ such that $q(i) \succ^SD_i p(i)$ for all $i \in N$ and $q(i) \succ^SD_i p(i)$ for some $i \in N$. An assignment $p$ is $SD$-efficient if there exists no assignment $q$ such that $q(i) \succ^SD_i p(i)$ for all $i \in N$ and $q(i) \succ^SD_i p(i)$ for some $i \in N$. An assignment is ex post efficient if it be can represented as a probability distribution over the set of Pareto optimal assignments. Perfection implies $SD$-efficiency which implies ex post efficiency.

An assignment rule is $SD$-efficient (ex post efficient) if it always returns an $SD$-efficient (ex post efficient) assignment. An assignment rule satisfies unanimity, if it returns the perfect assignment if a perfect assignment exists.

$SD$-efficiency implies ex post efficiency which implies unanimity. The first implication was shown by Bogomolnaia and Moulin [8]. For the second implication, assume that an assignment does not satisfy unanimity, there exists a perfect assignment $p$ but the mechanism returns some imperfect assignment $q$. The only Pareto optimal assignment is $p$. However since $q \neq p$, it cannot be achieved by a probability distribution over Pareto optimal discrete assignments.
Strategyproofness. A random assignment function $f$ is $SD$-strategyproof if $f(\succsim_i(i) \succeq_i^{SD} f(\succsim'_i, \succsim_{-i}))(i)$ for all $\succsim'_i \in \mathcal{R}(O)$. A random assignment function $f$ is weak $SD$-strategyproof if $-f(\succsim'_i, \succsim_{-i})(i) \succ_i^{SD} f(\succsim)(i)$ for all $\succsim'_i \in \mathcal{R}(O)$. It is easy to see that $SD$-strategyproofness implies weak $SD$-strategyproofness [8].

A random assignment function $f$ is weak $SD$-group-strategyproof if there never exists an $S \subset N$ and $\succsim'_S \in \mathcal{R}(O)^{|S|}$ such that $f(\succsim'_S, \succsim_{-S})(i) \succ_i^{SD} f(\succsim)(i)$ for all $i \in S$.

3. General impossibilities

For the random assignment problem for which the number of objects is not more than the number of agents, there exists a rule (PS) that is anonymous, neutral, $SD$-efficient and weak $SD$-strategyproof. However when the number of objects is more than the number of agents, we get the following impossibility theorem.

**Theorem 1.** For the assignment problem, there exists no anonymous, neutral, $SD$-efficient, and weak $SD$-strategyproof rule.

**Proof.** We consider a random assignment setting with two agents and four objects with the requirement that each agent gets two units of houses.

\begin{align*}
\succsim_1 & : a, b, c, d \\
\succsim_2 & : b, c, a, d \\
\succsim'_1 & : b, a, c, d \\
\succsim'_2 & : b, a, c, d
\end{align*}

Let us compute $f(\succsim_1, \succsim_2)$. By anonymity, and neutrality of $f$

\[ f(\succsim_1, \succsim'_2) = \begin{pmatrix} w & x & y & z \\ x & w & y & z \end{pmatrix}. \]

By $SD$-efficiency of $f$,

\[ f(\succsim_1, \succsim'_2) = \begin{pmatrix} 1 & 0 & y & z \\ 0 & 1 & y & z \end{pmatrix}. \]

By anonymity and neutrality of $f$,

\[ f(\succsim_1, \succsim'_2) = \begin{pmatrix} 1 & 0 & 1/2 & 1/2 \\ 0 & 1 & 1/2 & 1/2 \end{pmatrix}. \]

By using similar arguments, $SD$-efficiency, anonymity, and neutrality of $f$ implies that.

---

4The theorem requires an even weaker property than anonymity called equal treatment of equals that requires that agents with identical preferences should get the same allocation.
Now let us consider
\[
f(\succ_1, \succ_2) = \begin{pmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \end{pmatrix}.
\]
For \(f(\succ_1, \succ_2)\) to be feasible,
\[
x_{11}, x_{12}, x_{13}, x_{14}, x_{21}, x_{22}, x_{23}, x_{24} \geq 0
\]
\[
x_{11} + x_{12} + x_{13} + x_{14} = 2
\]
\[
x_{21} + x_{22} + x_{23} + x_{24} = 2
\]
\[
x_{11} + x_{21} = x_{12} + x_{22} = x_{13} + x_{23} = x_{14} + x_{24} = 1
\]

The first thing to notice is that since \(f(\succ_1, \succ_2)\) is \(SD\)-efficient, then either \(x_{21} = 0\) or \(x_{13} = 0\). If not, then agent 2 can give \(\epsilon\) fraction of \(a\) to agent 1 in return for \(\epsilon\) fraction of \(c\) and both agents benefit. This implies that \(f(\succ_1, \succ_2)\) is not \(SD\)-efficient.

Next, we show that if \(f(\succ_1, \succ_2) = f(\succ'_1, \succ_2)\) or \(f(\succ_1, \succ_2) = f(\succ_1, \succ'_2)\), then \(f\) is not weak \(SD\)-strategyproof.

If \(f(\succ_1, \succ_2) = f(\succ'_1, \succ_2)\), then
\[
f(\succ_1, \succ_2)(2) \succ_D f(\succ'_1, \succ_2)(2).
\]
Hence, \(f\) is not weak \(SD\)-strategyproof.

If \(f(\succ_1, \succ_2) = f(\succ_1, \succ'_2)\), then
\[
f(\succ'_1, \succ_2)(1) \succ_D f(\succ_1, \succ_2)(1).
\]
Hence, \(f\) is not weak \(SD\)-strategyproof.

Therefore the only way \(f\) can still be weak \(SD\)-strategyproof if both of the following conditions hold.

- \(f(\succ_1, \succ_2)(1)\) is incomparable for 1 with \(f(\succ'_1, \succ_2)(1)\).
- \(f(\succ_1, \succ_2)(2)\) is incomparable for 2 with \(f(\succ_1, \succ'_2)(2)\).

This means that the following constraints should hold.

(i) \(x_{11} + x_{12} > 1.5\) or \(x_{11} + x_{12} + x_{13} > 1.5\)

and

(ii) \(1 > x_{11}\) or
1.5 > x_{11} + x_{12}

and

(iii) \begin{itemize}
  \item x_{22} + x_{23} > 1.5 \text{ or }
  \item x_{22} + x_{23} + x_{21} > 1.5
\end{itemize}

and

(iv) \begin{itemize}
  \item 1 > x_{22} \text{ or }
  \item 1.5 > x_{21} + x_{22} + x_{23}
\end{itemize}

Since each of the conditions (i), (iv), (ii), (iii) need to be satisfied, it means that the following inequalities hold:

\begin{align*}
x_{11} + x_{12} + x_{13} &> 1.5 \\
x_{22} + x_{23} + x_{21} &> 1.5
\end{align*}

Adding both these inequalities yields

\begin{align*}
   x_{11} + x_{12} + x_{13} + x_{22} + x_{23} + x_{21} &> 3
\end{align*}

But this is a contradiction since

\begin{align*}
   x_{11} + x_{12} + x_{13} + x_{22} + x_{23} + x_{21} = (x_{11} + x_{21}) + (x_{12} + x_{22}) + (x_{13} + x_{23}) = 3.
\end{align*}

Hence if \( f \) is SD-efficient, neutral, and anonymous, then it cannot be weak SD-strategyproof.

The same argument can be extended to arbitrary number of agents where each agent requires two objects from among \( o_1, \ldots, o_{2n} \). Each new agent \( i \in \{2, \ldots, n\} \) most prefers objects \( o_{2i-1}, o_{2i} \) and least prefers objects \( o_1, o_2, o_3, o_4 \). Hence in each SD-efficient assignment each agent \( i \in \{2, \ldots, n\} \) is allocated \( o_{2i-1} \) and \( o_{2i} \) completely. The same arguments for the case of two agents apply to the more general case.

The proof above can be extended by cloning agents 1 and 2 to prove the following statement for the basic assignment setting.

**Theorem 2.** For the assignment problem, there exists no anonymous, neutral, SD-efficient, and weak SD weak group-strategyproofness rule even for equal number of agents and objects.

4. Multi-unit-eating PS

In this section, we examine the properties satisfied by multi-unit-eating PS (MPS). Before we proceed, we will try to get a better understanding of how multi-unit-eating PS works. Che and Kojima \( \text{[13]} \) defined multi-unit-eating PS as the rule in which each agent eats his \( c \) most preferred objects at speed 1 during the time interval \( t \in [0, 1] \). They assumed that at each point each agent has \( c \) objects available for consumption during the running of multi-unit-eating PS and hence all the objects are consumed at time 1. We first show that it may
be the case that less than \( c \) objects are available for consumption. Consider the illustration of multi-unit-eating PS in Figure 1. At time \( t = 7/8 \), only \( o_4 \) is remaining. Hence the first goal is to decide how to define multi-unit-eating PS when agents have less than \( c \) objects to eat. We resort to the following definition of multi-unit-eating PS.

Let \( \text{rem}(t) \) be the number of objects that have not been completely eaten at time \( t \). In multi-unit-eating PS, each agent eats his \( \max(c, \text{rem}(t)) \) favorite available objects with speed 1 at every time point until all the objects have been consumed.

![Figure 1: Illustration of multi-unit-eating PS with agents eating their preferred objects over time. The eventual assignment is \( p \).](image)

\[
\begin{array}{ccccccc}
\text{Agent 1} & o_1, o_2 & o_1, o_3 & o_1, o_4 & o_4 & \text{Agent 2} \\
& o_3, o_2 & o_3, o_4 & o_4 & o_4 & \text{0} & 1/2 & 3/4 & 7/8 & 1 & 9/8
\end{array}
\]

\[
p = \begin{pmatrix}
3/4 & 1/2 & 1/4 & 1/4 \\
1/4 & 1/2 & 3/4 & 3/4
\end{pmatrix}.
\]

We will use \( MPS \) as the abbreviation for multi-unit-eating PS. Our first observation is that even though agent may not necessarily eat \( c \) objects at each point, each agent eats the same number of objects.

**Observation 1.** At each time point, each agent is consuming the same number of objects. All the agents stop eating at exactly the same time.

If the number of objects is less than \( c \), then we know that only \( c' < c \) objects are remaining. Next, we study properties of multi-unit-eating PS. The first things to observe is that multi-unit-eating PS runs in linear time and results in a unique fractional assignment. We examine various axiomatic properties of multi-unit-eating PS. Our main findings are summarized in the following theorem. We will prove these properties in a series of propositions.

**Theorem 3.** Multi-unit-eating PS is single-valued, linear-time, SD envy-free, weak SD-strategyproof, and unanimous but not ex post efficient.
4.1. Fairness

We first show that multi-unit-eating PS satisfies all the notions of fairness defined in the preliminaries. It is easy to see that multi-unit-eating PS is anonymous and neutral. Next we show that multi-unit-eating PS is SD envy-free. For the proof, we use an extra bit of notation. For each set \( S \subseteq O \), let the characteristic vector of \( S \) be \((x_1, \ldots, x_m)\) where \( x_i = 1 \) if \( i \in S \) and \( x_i = 0 \) if \( i \notin S \).

**Proposition 1.** Multi-unit-eating PS is SD envy-free.

**Proof.** When multi-unit-eating PS is run, if at least one of the \( c \) most preferred available objects of some agent \( i \in N \) is finished, agent \( i \) starts eating the next \( c \) available objects. Also note that when an agent cannot consume more units of an object, then no agent can consume more units of the object either. We will refer to such a time-point as a breakpoint. The breakpoints are \( t_1, \ldots, t_l \). We prove by induction over the number of breakpoints in the algorithm, that for each agent \( i \in N \), his partial allocation \( p^k(i) \gtrsim_{SD} p^k(j) \) for all \( j \in N \).

For the base case \( k = 1 \), we know that \( p^1(i) \gtrsim_{SD} p^1(j) \) for all \( j \in N \) since each agent \( i \) was consuming his most preferred \( c \) objects. Now let us assume that \( p^k(i) \gtrsim_{SD} p^k(j) \). We show that \( p^{k+1}(i) \gtrsim_{SD} p^{k+1}(j) \). At time \( t^k \), let the number of objects that have not been completely even be \( c' \leq c \). Let us consider the time point \( t_k + \delta \) for some arbitrarily small \( \delta > 0 \). From time point \( t_k \) to \( t_k + \delta \) each agent \( i \) consumes \( \delta \) amount of \( c' \) most preferred objects of \( S \subseteq O \) for which \( \delta \) amount is still available. Thus \( p^k(i) \) is changed to \( p^k(i) + \delta(\hat{S}) \). In the meanwhile for each \( j \), \( p(j) \) is changed to \( p^k(j) + \delta(S') \) where \( S' \) consists of \( c' \) most preferred objects for which \( \delta \) amount is still available. Hence, \( p^{k+1}(i) \gtrsim_{SD} p^{k+1}(j) \) for each \( i, j \in N \).

**Corollary 1.** Multi-unit-eating PS is weak SD envy-free. Moreover, for the assignment problem without multi-unit demands, PS is SD envy-free.

4.2. Strategyproofness

In this subsection, we examine the strategic aspects of multi-unit-eating PS. We show that multi-unit-eating PS satisfies DL-strategyproofness and hence weak SD-strategyproofness. A random assignment function \( f \) is DL-strategyproof if \( f(\gtrsim(i)) \gtrsim^{DL}_{i} f(\gtrsim'(i)) \gtrsim_{-i}(i) \) for all \( \gtrsim \in \mathcal{R}(O) \).

**Lemma 1.** DL-strategyproofness implies weak SD-strategyproofness.

Next we show that multi-unit-eating PS is DL-strategyproof. The key to our argument is the insight that an agent cannot get an object with probability one if he does not start eating it from time \( t = 0 \). This contrasts sharply with one-at-a-time PS where an agent can still get an object completely even if he delays eating it.

**Proposition 2.** Multi-unit-eating PS is DL-strategyproof.
Proof. We show that for each agent \( i \in N \), \( MPS(N,O, (\succeq_i, \succeq_{-i}))(i) \gtrsim_{DL} MPS(N,O, (\succeq_i', \succeq_{-i}))(i) \) for all other preference \( \succeq_i' \in R(O) \) such that \( \succeq_i' \neq \succeq_i \). Without loss of generality, we assume that the preferences of agent \( i \) are as follows.

\[
i: \quad o_1, \ldots, o_m.
\]

Each agent eats his most preferred \( c \) objects at speed one during time period \([0,1]\). We first prove the following claim that if an agent does not immediately start eating one of his most preferred \( c \) object, then he cannot get it completely.

**Claim 1.** An agent cannot get an agent completely if he does not express it as one of his most preferred \( c \) objects.

**Proof.** Assume for contradiction that agent \( i \) starts eating \( o \) at time \( t > 0 \). Then this means that \( i \) finished eating \( o \) at time \( t > 1 \). But this means that all agents stopped eating at a time later than 1. If all agents had been eating their \( c \) most preferred available objects then the objects would have been finished at time 1. This means that at some time point before \( t = 1 \), there were less than \( c \) available objects. This implies that other agents also consumed some fraction of object \( o \). Hence agent \( i \) does not get \( o \) completely. Hence an agent can only get an object completely if he starts eating it at time 0.

We first show that an agent gets a less preferred allocation with respect to \( DL \) if he does not express his most preferred \( c \) objects as his most preferred \( c \) objects. Assume that an agent \( i \) gets his \( c \) most preferred objects completely. Then this is the best possible allocation for agent \( i \). Any other preference report by \( i \) in which he does not express most preferred \( c \) objects as his most preferred \( c \) objects. will yield a less preferred allocation with respect to \( DL \). Now assume that agent \( i \) gets \( o \) at least one of his most preferred \( c \) object partially. Then \( i \) only gets a less preferred allocation if \( i \) does not express \( o \) among his \( c \) most preferred alternatives. By delaying eating \( o \), \( i \) gets even less units of \( o \) which yields a less preferred allocation with respect to \( DL \). Hence we have established that in a \( DL \)-optimal preference of agent \( i \), the first \( c \) positions are taken by objects \( o_1, \ldots, o_c \).

Let us assume that in an agent’s optimal \( DL \) preference he truthfully expresses the first \( k \) objects in his preferences list. This is certainly the case for \( k \leq c \). We now prove by induction that in an agent’s optimal \( DL \) preference, he truthfully expresses his preferences over the first \( k + 1 \) objects. From Claim 1, we know that agent starts eating an object \( o_{k+1} \) at time \( t > 0 \), then \( o_{k+1} \) cannot be allocated completely to the agent. Hence if agent \( i \) has expressed objects \( o_1, \ldots, o_k \) in the optimal (truthful) order, then he should start eating it as soon as possible in order to maximize his share of \( o_{k+1} \). Assume that agent \( i \) does not express \( o_{k+1} \) as his \( k + 1 \) most preferred object and eats it later than he can while expressing \( o_1, \ldots, o_k \) as the initial part of his preference list. Let agent \( i \)’s misreport be \( \succeq_i' \). In that case agent \( i \) will get the same amount of objects in \( o_1, \ldots, o_k \) but strictly less fraction of \( o_{k+1} \). Hence \( MPS(N,O, (\succeq_i, \succeq_{-i}))(i) \gtrsim_{DL} MPS(N,O, (\succeq_i', \succeq_{-i}))(i) \).
Corollary 2. Multi-unit-eating PS is weak SD-SP.

As a corollary we also get that for \( m \leq n \), the original PS is weak SD-strategyproof. Our proof simplifies the argument in [Step 2, Proposition 1, §]. Note that Proposition 2 crucially depends on the fact that in MSP, each agent tries to eat his \( c \) most preferred objects. If each agent eats \( c - 1 \) most preferred objects, then we already know from [21], that the rule is then not even weak SD-strategyproof.

4.3. Efficiency

We now consider efficiency of multi-unit-eating PS. We first observe that multi-unit-eating PS satisfies unanimity.

Proposition 3. Multi-unit-eating PS satisfies unanimity.

Proof. A preference profile admits a perfect assignment only if each agent can his most preferred \( c \) objects. This implies that for any two agents, their sets of \( c \) most preferred objects don’t intersect. Given this condition, multi-unit-eating PS will assign each agent with his most preferred \( c \) objects.

Although unanimity is a very undemanding efficiency property, not all assignment rules satisfy unanimity. For example, the uniform assignment rule does not satisfy it. Even if multi-unit-eating PS is modified slightly so that agents eat their \( c + 1 \) most preferred objects at the same rate, then the modified rule would not satisfy unanimity. We also note that the allocation of each agent via multi-unit-eating PS is SD-preferred over the uniform allocation.

Proposition 4. For each agent \( i \in N \), \( i \) SD-prefers his allocation returned by multi-unit-eating PS to the uniform allocation.

Informally, an agent gets his worst possible assignment if all the other agents have the same preferences. Even in this case, each agent gets a uniform allocation. Although, multi-unit-eating PS satisfies unanimity, an assignment returned by multi-unit-eating PS can be represented as a convex combination of Pareto dominated discrete assignments.

Proposition 5. The outcome of multi-unit-eating PS can be represented as a probability distribution over Pareto dominated discrete assignments.

Proof. Consider two agents having the following preferences.

\[
1 : o_1, o_2, o_3, o_4 \\
2 : o_2, o_1, o_4, o_3
\]

The random assignment as a result of multi-unit-eating PS is

\[
\begin{pmatrix}
1/2 & 1/2 & 1/2 & 1/2 \\
1/2 & 1/2 & 1/2 & 1/2 \\
1/2 & 1/2 & 1/2 & 1/2
\end{pmatrix}
\]
which can be represented by a probability distribution over the following discrete assignments.

\[
\frac{1}{2} \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}.
\]

It can be shown that both discrete assignments are not Pareto optimal.

\[\square\]

**Corollary 3.** Multi-unit-eating PS is not SD-efficient.

**Proof.** An SD-efficient assignment cannot be represented as a convex combination of discrete assignment in which at least one of the assignments is not Pareto optimal. If this were the case, then the random assignment is not SD-efficient.

\[\square\]

Although the lack of SD-efficiency of multi-unit-eating PS was commented on in the original paper of Che and Kojima [13], we show that multi-unit-eating PS is surprisingly not even ex post efficient.

**Proposition 6.** Multi-unit-eating PS is not ex post efficient.

**Proof.** Consider two agents having the following preferences.

1: \(o_1, o_2, o_3, o_4\)
2: \(o_3, o_2, o_4, o_1\)

A discrete assignment is not Pareto optimal if agent 1 gets \(o_3\) or \(o_4\) and agent 2 gets \(o_1\). The only Pareto optimal discrete assignments are:

- \(\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}\)
- \(\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}\)
- \(\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}\)
- \(\begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\)
- \(\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}\)
- \(\begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}\).

We note that the outcome of multi-unit-eating PS is

\[
p = \begin{pmatrix} 7/8 & 4/8 & 2/8 & 3/8 \\ 1/8 & 4/8 & 6/8 & 5/8 \end{pmatrix}.
\]

Now if random assignment \(p\) is ex post efficient, then it can be expressed as a convex combination of Pareto optimal feasible assignments. Since \(p(2)(o_1) > 1\), this is only possible if \(\begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}\) is used in the convex combination. But since agent 2 does not get \(o_1\) in any other discrete permutation, this means that if any convex combination of Pareto optimal discrete assignments is used to obtain \(p\), then in each discrete Pareto optimal assignment used the following three cases can occur: (i) 2 gets both \(o_2\) and \(o_1\); (ii) 2 gets neither \(o_2\) nor \(o_1\) and (iii) 2 gets \(o_2\) but not \(o_1\). Hence, it must be that \(p(2)(o_2) \geq p(2)(o_1)\). But this is a contradiction.

\[\square\]
Table 1: Assignment rules for allocating multiple objects to agents with strict preferences.

| Property                  | Uniform | Priority | RP  | OPS  | MPS  |
|---------------------------|---------|----------|-----|------|------|
| SD-efficiency             |         | +        | -   | +    | -    |
| ex post efficient         |         | +        | +   | +    | -    |
| unanimity                |         | +        | +   | +    | +    |
| SD envy-freeness          | +       | -        | -   | +    | +    |
| weak SD envy-freeness     | +       | -        | +   | +    | +    |
| anonymous                 | +       | -        | +   | +    | +    |
| neutrality                | +       | +        | +   | +    | +    |
| SD-SP                     | +       | +        | +   | -    | -    |
| DL-SP                     | +       | +        | +   | -    | +    |
| weak SD-SP                | +       | +        | +   | -    | +    |
| polynomial-time           | +       | +        | -   | +    | +    |

Most of the properties of rules other than MPS (multi-unit-eating PS) are stated in [21].

5. Conclusions

In this paper, we showed a general impossibility result concerning randomized assignment with multi-unit demands. Another impossibility result requiring weak SD-group-strategyproofness applies to randomized assignment without multi-unit demands. We then presented a definition of multi-unit-eating PS. Multi-unit-eating PS has previously only been defined inaccurately in the literature. We showed that whereas multi-unit-eating PS satisfies some compelling fairness and strategic properties, it does not satisfy reasonable efficiency requirements.

\[
(SD\text{-envy-free} \& \text{weak } SD\text{-SP}) \text{ MPS} \quad OPS \quad (SD \text{ envy-free} \& SD\text{-efficient})
\]

\[
\text{PS} \quad (SD \text{ envy-free, weak } SD\text{-SP} \& SD\text{-efficient})
\]

Figure 2: MPS and OPS are two extensions of PS for multi-unit demands. Both are equivalent to PS under single-unit demand. Under multi-unit demands, whereas OPS is not weak SD-strategyproof, MPS is not SD-efficient.

Our findings concerning multi-unit-eating PS are summarized in Table 1 which also provides a comparison with other random assignment rules. In view of the impossibility result (Theorem 1), it is not possible to achieve the desirable properties of PS and multi-unit-eating PS simultaneously. It is easy to see that the choice of an assignment rule depends on which properties are prioritized (see Figure 2). We leave a characterization of multi-unit-eating PS for future work.
Acknowledgments

This material is based upon work supported by the Australian Government’s Department of Broadband, Communications and the Digital Economy, the Australian Research Council, the Asian Office of Aerospace Research and Development through grant AOARD-124056.

References

[1] Abdulkadiroğlu, A., Sönmez, T., 1998. Random serial dictatorship and the core from random endowments in house allocation problems. Econometrica 66 (3), 689–702.

[2] Abdulkadiroğlu, A., Sönmez, T., 1999. House allocation with existing tenants. Journal of Economic Theory 88 (2), 233–260.

[3] Abraham, D. J., Cechlárová, K., Manlove, D., Mehlhorn, K., 2005. Pareto optimality in house allocation problems. In: Proceedings of the 16th International Symposium on Algorithms and Computation (ISAAC). Vol. 3341 of Lecture Notes in Computer Science (LNCS). pp. 1163–1175.

[4] Athanassoglou, S., Sethuraman, J., 2011. House allocation with fractional endowments. International Journal of Game Theory 40 (3), 481–513.

[5] Aziz, H., Brandt, F., Brill, M., 2013. The computational complexity of random serial dictatorship. Economics Letters 121 (3), 341–345.

[6] Aziz, H., Brandt, F., Stursberg, P., 2013. On popular random assignments. In: Vöcking, B. (Ed.), Proceedings of the 6th International Symposium on Algorithmic Game Theory (SAGT). Vol. 8146 of Lecture Notes in Computer Science (LNCS). Springer-Verlag, pp. 183–194.

[7] Bhalgat, A., Chakrabarty, D., Khanna, S., 2011. Social welfare in one-sided matching markets without money. In: Proceedings of APPROX-RANDOM. pp. 87–98.

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

[9] Bouveret, S., Endriss, U., Lang, J., 2010. Fair division under ordinal preferences: Computing envy-free allocations of indivisible goods. In: Proceedings of the 19th European Conference on Artificial Intelligence (ECAI). pp. 387–392.

[10] Bouveret, S., Lang, J., 2011. A general elicitation-free protocol for allocating indivisible goods. In: Proceedings of the 22 International Joint Conference on Artificial Intelligence (IJCAI). pp. 73–78.
[11] Budish, E., Che, Y.-K., Kojima, F., Milgrom, P., 2013. Designing random allocation mechanisms: Theory and applications. American Economic Review Forthcoming.

[12] Chambers, C., 2004. Consistency in the probabilistic assignment model. Journal of Mathematical Economics 40, 953–962.

[13] Che, Y.-K., Kojima, F., 2010. Asymptotic equivalence of probabilistic serial and random priority mechanisms. Econometrica 78 (5), 1625—1672.

[14] Crès, H., Moulin, H., 2001. Scheduling with opting out: Improving upon random priority. Operations Research 49 (4), 565–577.

[15] Gärdenfors, P., 1973. Assignment problem based on ordinal preferences. Management Science 20, 331–340.

[16] Guo, M., Conitzer, V., 2010. Strategy-proof allocation of multiple items without payments or priors. In: Proceedings of the 9th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS). pp. 881–888.

[17] Heo, E. J., 2011. Probabilistic assignment with multiple demands: A generalization and a characterization of the serial rule. Tech. Rep. 1809195, SSRN.

[18] Hylland, A., Zeckhauser, R., 1979. The efficient allocation of individuals to positions. The Journal of Political Economy 87 (2), 293–314.

[19] Kalinowski, T., Narodytska, N., Walsh, T., Xia, L., 2013. Strategic behavior when allocating indivisible goods sequentially. In: Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI).

[20] Katta, A.-K., Sethuraman, J., 2006. A solution to the random assignment problem on the full preference domain. Journal of Economic Theory 131 (1), 231–250.

[21] Kojima, F., 2009. Random assignment of multiple indivisible objects. Mathematical Social Sciences 57 (1), 134—142.

[22] Svensson, L.-G., 1994. Queue allocation of indivisible goods. Social Choice and Welfare 11, 323–330.

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

[24] Wilson, L., 1977. Assignment using choice lists. Operations Research Quarterly 28 (3), 569—578.

[25] Yilmaz, O., 2009. Random assignment under weak preferences. Games and Economic Behavior 66 (1), 546–558.

[26] Young, H. P., 1995. Dividing the indivisible. American Behavioral Scientist 38, 904–920.
Appendix

Proof of Theorem

Proof. We consider a random assignment setting with four agents and four objects. There are two agents that are of type 1 and two agents of type 2. Let the real preferences of the agents \( \{1, 2\} \) of type 1 be \( \succsim_1 \) and let the real preferences of agents \( \{3, 4\} \) of type 2 be \( \succsim_2 \).

\[
\begin{align*}
\succsim_1 & : a, b, c, d \\
\succsim_2 & : b, c, a, d \\
\succsim'_1 & : b, a, c, d \\
\succsim'_2 & : b, a, c, d
\end{align*}
\]

Let us compute \( f(\succsim_1, \succsim_1, \succsim'_2, \succsim'_2) \).

By anonymity, and neutrality, we know that

\[
f(\succsim_1, \succsim_1, \succsim'_2, \succsim'_2) = \begin{pmatrix}
w/2 & x/2 & y/2 & z/2 \\
w/2 & x/2 & y/2 & z/2 \\
x/2 & w/2 & y/2 & z/2 \\
x/2 & w/2 & y/2 & z/2
\end{pmatrix}.
\]

By \( SD \)-efficiency, we know that

\[
f(\succsim_1, \succsim_1, \succsim'_2, \succsim'_2) = \begin{pmatrix}
1/2 & 0 & y/2 & z/2 \\
1/2 & 0 & y/2 & z/2 \\
0 & 1/2 & y/2 & z/2 \\
0 & 1/2 & y/2 & z/2
\end{pmatrix}.
\]

Due to anonymity and neutrality of \( f \),

\[
f(\succsim_1, \succsim_1, \succsim'_2, \succsim'_2) = \begin{pmatrix}
1/2 & 0 & 1/4 & 1/4 \\
1/2 & 0 & 1/4 & 1/4 \\
0 & 1/2 & 1/4 & 1/4 \\
0 & 1/2 & 1/4 & 1/4
\end{pmatrix}.
\]

By using similar arguments, \( SD \)-efficiency, anonymity, and neutrality of \( f \) implies that

\[
f(\succsim'_1, \succsim'_1, \succsim'_2, \succsim_2) = \begin{pmatrix}
1/2 & 1/4 & 0 & 1/4 \\
1/2 & 1/4 & 0 & 1/4 \\
0 & 1/4 & 1/2 & 1/4 \\
0 & 1/4 & 1/2 & 1/4
\end{pmatrix}.
\]

Now let us consider

16
both of the following conditions hold. Hence, 
\[ f(\tilde{z}_1, \tilde{z}_1, \tilde{z}_2, \tilde{z}_2) = \begin{pmatrix} x_{11}/2 & x_{12}/2 & x_{13}/2 & x_{14}/2 \\ x_{11}/2 & x_{12}/2 & x_{13}/2 & x_{14}/2 \\ x_{21}/2 & x_{22}/2 & x_{23}/2 & x_{24}/2 \\ x_{21}/2 & x_{22}/2 & x_{23}/2 & x_{24}/2 \end{pmatrix}, \]

For \( f(\tilde{z}_1, \tilde{z}_1, \tilde{z}_2, \tilde{z}_2) \) to be feasible,

\[
x_{11}, x_{12}, x_{13}, x_{14}, x_{21}, x_{22}, x_{23}, x_{24} \geq 0
\]
\[
x_{11} + x_{12} + x_{13} + x_{14} = 2
\]
\[
x_{21} + x_{22} + x_{23} + x_{24} = 2
\]
\[
x_{11} + x_{21} = x_{12} + x_{22} = x_{13} + x_{23} = x_{14} + x_{24} = 1
\]

The first thing to notice is that since \( f(\tilde{z}_1, \tilde{z}_1, \tilde{z}_2, \tilde{z}_2) \) is \( SD \)-efficient, then either \( x_{21} = 0 \) or \( x_{13} = 0 \). If not, then agents 2 can give \( \epsilon \) fraction of \( a \) to agent 1 in return for \( \epsilon \) fraction of \( c \) and all agents benefit. This implies that \( f(\tilde{z}_1, \tilde{z}_1, \tilde{z}_2, \tilde{z}_2) \) is not \( SD \)-efficient.

Next, we show that if \( f(\tilde{z}_1, \tilde{z}_1, \tilde{z}_2, \tilde{z}_2) = f(\tilde{z}_1, \tilde{z}_1, \tilde{z}_2, \tilde{z}_2) \) or \( f(\tilde{z}_1, \tilde{z}_1, \tilde{z}_2, \tilde{z}_2) = f(\tilde{z}_1, \tilde{z}_1, \tilde{z}_2, \tilde{z}_2) \), then \( f \) is not weak \( SD \) weak group-strategyproof.

If \( f(\tilde{z}_1, \tilde{z}_1, \tilde{z}_2, \tilde{z}_2) = f(\tilde{z}_1, \tilde{z}_1, \tilde{z}_2, \tilde{z}_2) \), then

\[
f(\tilde{z}_1, \tilde{z}_1, \tilde{z}_2, \tilde{z}_2)(3) \gneq SD f(\tilde{z}_1, \tilde{z}_1, \tilde{z}_2, \tilde{z}_2)(3).
\]
Hence, \( f \) is not weak \( SD \)-weak group-strategyproof.

If \( f(\tilde{z}_1, \tilde{z}_1, \tilde{z}_2, \tilde{z}_2) = f(\tilde{z}_1, \tilde{z}_1, \tilde{z}_2, \tilde{z}_2) \), then

\[
f(\tilde{z}_1, \tilde{z}_1, \tilde{z}_2, \tilde{z}_2)(1) \gneq SD f(\tilde{z}_1, \tilde{z}_1, \tilde{z}_2, \tilde{z}_2)(1).
\]
Hence, \( f \) is not weak \( SD \)-weak group-strategyproof.

Therefore the only way \( f \) can still be weak \( SD \)-weak group-strategyproof if both of the following conditions hold.

- \( f(\tilde{z}_1, \tilde{z}_1, \tilde{z}_2, \tilde{z}_2)(1) \) is incomparable for type 1 with \( f(\tilde{z}_1, \tilde{z}_1, \tilde{z}_2, \tilde{z}_2)(1) \).
- \( f(\tilde{z}_1, \tilde{z}_1, \tilde{z}_2, \tilde{z}_2)(3) \) is incomparable for type 2 with \( f(\tilde{z}_1, \tilde{z}_1, \tilde{z}_2, \tilde{z}_2)(3) \).

This means that the following constraints should hold.

(i) \( x_{11} + x_{12} > 1.5 \) or \( x_{11} + x_{12} + x_{13} > 1.5 \)

and

(ii) \( x_{22} + x_{23} > 1.5 \) or \( x_{22} + x_{23} + x_{21} > 1.5 \)
This means that the following inequalities hold:

\[ x_{11} + x_{12} + x_{13} > 1.5 \]
\[ x_{22} + x_{23} + x_{21} > 1.5 \]

Hence,

\[ x_{11} + x_{12} + x_{13} + x_{22} + x_{23} + x_{21} > 3 \]

But this is a contradiction since

\[ x_{11} + x_{12} + x_{13} + x_{22} + x_{23} + x_{21} = (x_{11} + x_{21}) + (x_{12} + x_{22}) + (x_{13} + x_{23}) = 3. \]

Hence if \( f \) is \( SD \)-efficient, neutral, and anonymous, then it cannot be weak \( SD \)-weak group-strategyproof.

The same argument can be extended to arbitrary number of agents. \( \square \)