Random Assignment: Redefining the Serial Rule

Anna Bogomolnaia

*Rice University* → *University of Glasgow*

June 2013
Model:

\[ A = \{a_1, \ldots, a_m\} \text{ — set of indivisible objects} \]
\[ N = \{1, \ldots, n\} \text{ — set of agents} \]

Each agent wants exactly one object and has ordinal preferences \( R_i \in \mathcal{R} \) over \( A \)
\( \mathcal{R} \) is the set of all (strict or non-strict) orderings of \( A \)

Mechanism design: no money, but randomization is allowed
Agents are only asked about their ordinal rankings \( R_i \) (private information)

A probabilistic assignment is an \( |N| \times |A| \) matrix \( P = (p_{ia})_{i \in N \ a \in A} \)

Interpretation: \( p_{ia} \) is the probability that agent \( i \) gets object \( a \).

Agents have cardinal preferences and use expected utility to compare lotteries

Alternative interpretation: objects are infinitely divisible, each agent is entitled to exactly one unit total of objects; \( p_{ia} \) is the share of object \( a \) that agent \( i \) gets.

Preferences are linear in quantity for each object and are additive across objects.

The model is easily extendable to the case where some objects can have several copies, or even be available in a non-integer amount.
A **probabilistic assignment** is an \(|N| \times |A|\) matrix \(P = (p_{ia})_{i \in N, a \in A}\). Here \(p_{ia}\) is the share/probability of object \(a\) allotted to agent \(i\).

Notation: \(P_i\) is the row of agent \(i\)’s probabilities to receive different goods (“agent \(i\)’s assignment at \(P\)”) \(P^a\) is the column, specifying chances of good \(a\) to be given to different agents.

A **deterministic assignment** is a permutation matrix, i.e., \(n \times n\) zero-one matrix with exactly one “1” in each row and each column. A bistochastic matrix: one with nonnegative entries, and with sums in all columns and rows are equal to one. Fact: any bistochastic matrix is a convex combination of permutation matrices. Thus, the set \(\mathcal{P}\) of feasible probabilistic assignments is the set of all \(n \times n\) bistochastic matrices.

A (probabilistic, ordinal) assignment rule is a correspondence \(f: \mathcal{R}^n \rightrightarrows \mathcal{P}\) which is essentially single-valued: \(f(R) \subset \mathcal{P}, f(R) \neq \emptyset\), and all agents are indifferent between any two \(P, P' \in f(R)\).
Examples of probabilistic rules (definitions for strict domain)

**Random priority rule** orders agents using uniform lottery, and then lets them pick their most preferred objects in that order.

**Serial rule** $S$ defines random assignments using “simultaneous eating” process. Agents acquire probabilities of objects continuously over the unit interval of time $[0; 1]$, simultaneously and at the same unit rate.

Fix a preference profile $R \in \mathcal{R}^n$.

Each agent starts by consuming his most preferred object. When the object he consumes is exhausted, he switches to his next most preferred object among the still available ones.

At any moment $\tau \in [0, 1]$, we have $S[\tau] = S(R)[\tau]$, a partial serial assignment that agents obtained until $\tau$. The final assignment $S = S[1]$ is given by vectors of probabilities agents consumed along this process.

Random priority fairs better on incentives, Serial rule fairs better on efficiency and fairness

**Our goal:** alternative definition (or characterization) of Serial rule, good for both strict and full domain
Related literature:

Bogomolnaia and Moulin (2001) introduced the Serial rule on the strict domain, and showed that it satisfies sd efficiency and sd no-envy.

Katta and Sethuraman (2006) extended Serial rule to the full domain (a complicated definition, based on the repeated application of finding a maximal flow on a network algorithm).

Characterizations of Serial rule (all in 2011-2012):
On the strict domain: Heo (2011); Kesten, Kurino, and Unver (2011); Hashimoto and Hirata (2011), Bogomolnaia and Heo (2011), Hashimoto, Hirata, Kesten, Kurino, and Unver (2011)
On the full domain: Heo and Yilmaz (2012)
Essentially all those results are by means of sd efficiency, sd no-envy, and a third, invariance-type requirement. Very complicated proofs (especially for the most weak invariance-type axiom).

We provide an alternative, welfarist axiomatization, or re-definition of the Serial rule, which works equally well for both strict and non-strict preferences.
Representation of any assignment as the result of “consumption process”:
Let $R \in \mathcal{R}^N$ be a preference profile and $P \in \mathcal{P}$ be a feasible probabilistic assignment. We interpret $P$ as the output of continuous consumption process over time interval $[0, 1]$. Each agent $i$ consumes objects with unit speed and in decreasing order of his preferences. Contrary to Serial rule, agent $i$ consumes exactly the share $p_{ia}$ of each object $a$, and then switches to the next object in his order.
We fix an arbitrary order of objects $a_1, \ldots, a_m$, and assume that when an agent is indifferent between several objects he consumes first those which come earlier in this ordering.
We define $P[\tau]$ to be the partial assignment matrix by time $\tau \in [0, 1]$.
Given $R \in \mathcal{R}^N$, Serial rule $S$ is defined similarly. We do not have a “final assignment” which pre-specifies objects’ shares. Instead, agents eat goods until they are exhausted, in decreasing order of their preferences. However, given $S(R)$, we have that $S(R)[\tau]$ is exactly its partial assignment by time $\tau$, as defined above.
Consumption process $(P[\tau])_{\tau \in [0,1]}$ is continuous in $\tau$, and piece-wise linear: it has only finite number of “switches” (at most $n(n - 1)$), when some agent changes from one object to another.
Fix a preference profile $R$.

$U_i(a) = U(R_i, a) = \{o \in A : oR_ia\}$ (the upper contour set of $a$ under preferences $R_i$)

$U_i^k$ — the set of all objects in the top $k$ indifference classes for the preferences $R_i$

Given the assignment $P$, we define $t_i(a) = t_i^P(a) = \sum_{b \in U_i(a)} p_{ib}$, the total share of objects at least as good as $a$, which agent $i$ receives under $P$.

$t_i^P(a)$ is exactly the last moment in the consumption process $P[\tau]$ when agent $i$ consumes objects at least as good as $a$.

For each agent $i$, we define a vector $t_i = (t_i(1), \ldots, t_i(K))$, where $K$ is the number of indifference classes in $R_i$. Here $t_i(k) = \sum_{b \in U_i^k} p_{ib}$ is the total share of objects agent $i$ gets from his top $k$ indifference classes.

**Theorem 1** For any $R$, Serial assignment $S(R)$ is the unique one which lexicographically maximizes the vector of shares $(t_1, \ldots, t_n)$.

**Note:** For any $x \in \mathbb{R}^d$, let $x^*$ be a permutation of coordinates of $x$ in the increasing order: $x_1^* \leq x_2^* \leq \ldots \leq x_d^*$. We say that $xLy$ ($x$ is lexicographically preferred to $y$), if there is a $j \in \{1, \ldots, d\}$ such that $x_j^* > y_j^*$, while $x_i^* = y_i^*$ for all $i < j$. 
**Proof** (strict preference profile $R$)

For a strict profile $R$, the vector $(t_1, \ldots, t_n)$ coincides with the vector of $t_i(a) = \sum_{b \in U_i(a)} p_{ib}$, for all $i \in N$, $a \in A$.

Let $P \neq S$, and let $\tau \in [0, 1]$ be the last moment $t$ such that $P[t] = S[t]$.

There is an agent $j$ and an object $b$, such that $b$ is still available at time $\tau$, but under $P[t]$ this agent $j$ switches at the moment $\tau$ to some object $c$ he values less than $b$.

Note that $t_i^S(a) = t_i^P(a)$ for all $i$, $a$ such that $t_i^S(a) \leq \tau$, while $t_i^P(a) > \tau$ implies $t_i^S(a) > \tau$.

However, $t_j^S(b) > \tau$ while $t_j^P(b) = \tau$. 
Defining of Serial Rule for non-strict preferences (Algorithm)

(1) Set $A_1 = A, \ c(i, 1) = 0$ for all $i \in N, \ k = 1$

(2) Step $k$: At each step $k$, we find the largest share $\lambda_k$ such that each agent can consume at least $\lambda_k$ of her most desired objects from $A_k$, the set of objects still available at this step. Each agent is then guaranteed the share $\lambda_k$ of her preferred objects. Maximality of $\lambda_k$ implies that some objects are exhausted at this point. We define $A_{k+1}$ to be the set of not yet exhausted objects and proceed to step $k + 1$.

For this purpose, we construct the following directed network, with the set of ages $A_k \cup N \cup \{s\} \cup \{t\}$. (i) From the “source” $s$ we draw an arc of capacity $c(i, k) + \lambda$ toward each agent $i \in N$. (ii) From each agent $i \in N$ we draw an arc of infinite capacity toward each of her best in $A_k$ objects (iii) From each object in $A_k$ we draw an arc of capacity 1 toward the “sink” $t$.

Here $\lambda \geq 0$ is a parameter. For each $\lambda$ we find the maximal flow through this network, which can be sent from the source $s$ to the sink $t$. When we continuously increase $\lambda$ from zero on, there is the last moment $\lambda = \lambda_k$ after which the maximal flow is less then the total “out” capacity of the source, $\sum_{i \in N} (c(i, k) + \lambda)$. 
Maximal flow through a network is known to be equal to the capacity of a minimal cut. A cut is a partition of the network’s nodes into $S_s \ni s$ and $S_t \ni t$, and its capacity is the sum of capacities of all arcs going from $S_s$ to $S_t$. In our network, all cuts of finite capacity are such that $S_s = \{s\} \cup X \cup W$ where $X \subset N$ and $M_X(A_k) \subset W \subset A_k$. The capacity of such cut is $\sum_{i \in N \setminus X} (c(i, k) + \lambda) + |W|$, so in a minimal cut it has to be $M_X(A_k) = W$.

When $\lambda \leq \lambda_k$, the maximal flow is equal to the “out” capacity of the source ($S_s = \{s\}$ is a minimal cut). For $\lambda > \lambda_k$, $S_s = \{s\}$ stops to be a minimal cut. Hence (by continuity of our process in $\lambda$), when $\lambda = \lambda_k$, there is another minimal cut with $S_s = \{s\} \cup X_k$ where $X_k \subset N$. It has to be that $X_k \in \arg\min_{X \subset N} \left( \sum_{i \in N \setminus X} (c(i, k) + \lambda) + |M_{X_k}(A_k)| \right)$. If the above $\arg\min$ is not a singleton, we choose $X_k$ to be the largest set in the sense of inclusion$^1$.

---

$^1$ It is easy to check that a union of two such “minimal” $X$ will also be minimal.
We hence must have \[ \sum_{i \in N} (c(i, k) + \lambda_k) = \sum_{i \in N \setminus X_k} (c(i, k) + \lambda_k) + |M_{X_k}(A_k)|, \]
or

\[
\lambda_k = \frac{|M_{X_k}(A_k)| - \sum_{i \in X_k} c(i, k)}{|X_k|} = \min_{X \subset N} \frac{|M_X(A_k)| - \sum_{i \in X} c(i, k)}{|X|}.
\]

Agents from \( X_k \) constitute the “bottleneck” of our algorithm at Step \( k \). When each agent from \( X_k \) is assigned her “in” capacity \( c(i, k) + \lambda_k \) of her top good, the set \( M_{X_k}(A_k) \) of their top goods is completely exhausted. Maximality of \( X_k \) implies that it is feasible to give all agents from \( N \setminus X_k \) shares, strictly larger then their “in” capacity \( c(i, k) + \lambda_k \), of their top in \( A_k \) goods, using only goods from \( A_k \setminus M_{X_k}(A_k) \).

Using this observation, we assign to each agent \( i \in X_k \) share \( \lambda_k \) of her top in \( A_k \) goods. We define the set of available goods for the next step, \( A_{k+1} = A_k \setminus M_{X_k}(A_k) \). Finally, we set the new capacity of agents from the bottleneck, \( i \in X \), to be \( c(i, k + 1) = 0 \), and we increase the capacity of agents, not involved in the bottleneck, \( j \in N \setminus X_k \), to be \( c(j, k + 1) = c(j, k) + \lambda_k \).
Since at each step at least one good is exhausted, the algorithm will finish in $K \leq m$ steps, with each agent getting exactly one unit total of goods’ shares, and with $\sum_{1 \leq k \leq K} \lambda_k = 1$. Note that there could be multiple maximal flows (and hence multiple assignments) at each step, but all of them are utility-equivalent.
Proof of the Theorem 1 (non-strict preferences)

Fix an arbitrary preference profile \( R \). Let \( P \notin S(R) \) and let \( \tau \in [0, 1] \) be the last moment \( t \) such that \( P[t] \in \{S[t] : S \in S(R)\} \). Let \( r \) be such that \( \sum_{1 \leq k \leq r} \lambda_k \leq \tau < \sum_{1 \leq k \leq r+1} \lambda_k \).

Thus, the consumption process \( P[t] \) is done as if in Serial rule algorithm during its first \( r \) steps (in both assigned and guaranteed shares), but deviates from the Serial assignment algorithm during its step \( r + 1 \).

In (any) Serial consumption process, along the time interval \( \left[ \sum_{1 \leq k \leq r} \lambda_k, \sum_{1 \leq k \leq r+1} \lambda_k \right] \) everybody consumes her best goods from \( A_{r+1} \). However, the consumption process \( P[t] \) differs. Thus, as in the case of strict preferences, there is an agent \( j \) and an object \( b \), such that the object \( b \) is still available at time \( \tau \), but under \( P[t] \) this agent \( j \) switches at the moment \( \tau \) to some object \( c \) which she values strictly less then \( b \). Let object \( b \) be from the \( l \)-th indifference class for this agent \( j \). Again, \( t^S_i(k) = t^P_i(k) \) for all \( i \) and their indifference classes \( k \), such that \( t^S_i(k) \leq \tau \), while \( t^P_i(k) > \tau \) implies \( t^S_i(k) > \tau \). However, \( t^S_i(l) > \tau \) while \( t^P_i(l) = \tau \).
Theorem 2
Serial rule is the only one which is sd-efficient, sd-envy-free, and strategy-proof on the lexicographic preference domain.

Proof.
Serial rule is well-known to be sd-efficient and sd-envy-free. Shulman and Vazirani (2012) shows that Serial rule is strategy-proof on the lexicographic preference domain. HHKKU (2012) shows that Serial rule is the only one which is sd-efficient, sd-envy-free, and satisfies limited invariance. Thus, it is enough to show that strategy-proofness on the lexicographic domain implies limited invariance.

Limited Invariance: rule $P(R)$ satisfies it, if the following is true for any initial preference profile $R$, $a \in A$, and $i \in N$. Whenever agent $i$ changes her ordering of objects she ranks below $a$, it does not affect her share of object $a$. Formally, if $R'_i$ coincides with $R_i$ on $U(R'_i, a) = U(R_i, a)$, then $p_{ia}(R) = p_{ia}(R'_i, R_{-i})$.
Indeed, assume that a rule $f$ fails limited invariance. Hence, there is a preference profile $R$, agent $i$, object $a$, and preferences $R'_i$, with $U(R'_i, a) = U(R_i, a)$ and $R'_i|U(R_i, a) = R_i|U(R_i, a)$ and , such that $\sum_{b \in U(R_i, a)} p_{ib}(R) \neq \sum_{b \in U(R_i, a)} p_{ib}(R_i, R_{-i})$. Given such $R$ and $i$, assume $a$ to be the best for agent $i$ object among those for which this inequality is true. Then, for any $c$ which is better for $i$ then $a$ (at either $R_i$ or $R'_i$), we have $\sum_{b \in U(R_i, c)} p_{ib}(R) =$ $\sum_{b \in U(R_i, c)} p_{ib}(R_i, R_{-i})$. Hence, for those $c$ we have $p_{ic}(R) = p_{ic}(R_i, R_{-i})$, while for $a$ we obtain $p_{ic}(R) \neq p_{ic}(R_i, R_{-i})$.

Thus, there is a manipulation for agent $i$ (either at profile $R$, or at profile $R' = (R_i, R_{-i})$) which increases his total share of good $a$, keeping the same his shares of goods better then $a$. The result of this manipulation will be lexicographically preferred to telling the truth, so strategy-proofness is violated.