New descriptions of serial dictatorship for object allocation with indifferences

Will Sandholtz* Andrew Tai†

October 2024

Abstract

Serial dictatorship (SD) is often used to allocate indivisible objects to participants. However, when participants may be indifferent between objects, the usual implementation is not Pareto efficient. We note the correct implementation of SD, which acts on social outcomes (allocations). We also note two other descriptions of the same mechanism, which do not require participants to choose between social outcomes.

1 Introduction

We consider allocating indivisible objects to agents where preferences may contain indifferences. In the classically laid out serial dictatorship, the first priority agent selects his most preferred object, with indifferences broken arbitrarily. The next agent selects his most preferred object among those remaining, indifferences again broken arbitrarily. We term this "naive serial dictatorship" (NSD). However, this mechanism is not Pareto efficient, as will be shown in the next section.

In practice, NSD is used in important settings where indifferences are plausible. For example, US Naval Academy graduating midshipmen select their first ship assignments this way.[1] A midshipman may plausibly be indifferent between two ships of the same class, or two different ships based in the same location. Other plausible examples are NYC Specialized High Schools matching and college dorm assignments.

In this note, we present the "correct" implementation of serial dictatorship for this setting. While this mechanism is known, to our best knowledge it has not been well recorded. The mechanism solicits each agent's preferences over the objects. However, instead of straight forwardly allocating one of the favorite remaining objects to the n^{th} agent, the mechanism preserves all *allocations* in which n matches to one of his favorite remaining objects. Equivalently, it vetoes any allocation in which he does not receive one of these objects. That is, n is guaranteed to match with one of his favorite remaining objects, though the exact one may not determined until later. We refer to this as "true serial dictatorship" (TSD) for convenience to distinguish it from NSD.

While this description of TSD is straight-forward for those with some training in social choice, it can be difficult to describe to participants and market stakeholders. Thus we give two other descriptions of TSD. The first has the market implementer check compatible allocations to allow participants to select sets of objects.

*UC Berkeley, willsandholtz@berkeley.edu

†Defense Resources Management Institute, US DoD, andrew.tai@nps.edu

[1] For a video of the process, see here: https://www.youtube.com/watch?v=1nvrzluVIeQ

The second description deals almost almost entirely with objects themselves rather than allocations. These descriptions can be useful when the market stakeholders are not familiar with social choice.

TSD preserves strategyproofness and Pareto efficiency. However, neither TSD nor NSD are group strategyproof. Pápai (2003) shows that with strict preferences, group strategyproofness is equivalent to non-bossiness and strategyproofness. However, this is not true on the full domain of preferences. Indeed, Ehlers (2002) shows that Pareto efficiency and group strategyproofness are incompatible when agents have indifferences. Thus we recover as much as possible the desirable characteristics of serial dictatorship from strict preferences. Of course, in the full domain of preferences, the house allocation problem is bound to the Arrow (2012) Impossibility Theorem. No rule can satisfy non-dictatorship, Pareto efficiency, and independence of irrelevant alternatives.

Table 1: Comparison of mechanisms

	Naive SD	True SD
Pareto efficient	No	Yes
Strategyproof	Yes	Yes
Group strategyproof	No	No

In the next section, we formally define the model. In Section 3, we describe NSD and TSD. In Section 4, we prove the desired properties of TSD and illustrate why it is not group strategyproof.

2 Model

We briefly describe the model, which is standard. Let the set of agents be $N = \{1, ..., n\}$ and indivisible objects be $H = \{1, ..., h\}$. We do not impose any relation between $|N|$ and $|H|$. Each $i \in N$ has a preference relation R_i over $H \cup \{\emptyset\}$ which may include indifferences, where $\emptyset$ represents being unmatched. If $|N| \leq |H|$, $\emptyset$ may be omitted in the absence of an outside option. We denote I_i the indifference relation from R_i, and P_i the strict relation. A preference profile is $R = (R_i)_{i \in N}$. For convenience, for R_i we denote R_{-i} to be the preference profile of other agents $N \setminus \{i\}$. For $C \subseteq N$, R_C is the profile of R_i such that $i \in C$; likewise, R_{-C} is the profile of $N \setminus C$.

An **allocation** is a vector $x = (x_1, ..., x_n)$ where each $h \in H$ corresponds to at most one x_i. That is, each object is assigned to at most one agent. Let X be the set of all feasible allocations. An **allocation rule** f is a function that assigns any preference profile R to an allocation, denoted $f(R) = x$. The desiderata are as follows.

Definition 1. An allocation rule f is **Pareto efficient** if for any R, $f(R)$ is Pareto efficient according to R. That is, f only selects allocations that are Pareto efficient.

Definition 2. An allocation rule f is **strategyproof** if for all $i \in N$, for any R_i, R'_i, R_{-i}, we have $f_i(R_i, R_{-i}) R_i f_i(R'_i, R_{-i})$. Informally, if i changes his report to R'_i, he is weakly worse off.

Definition 3. An allocation rule f is **group strategyproof** if for all $C \subseteq N$ and for any R_C, R'_C, R_{-C}, there is some $i \in N'$ such that $f_i(R_N, R_{-N}) P_i f_i(R'_N, R_{-N})$. Informally, for any coalition misreport, at least one member of the coalition is strictly worse off.

3 Allocation rules

We now specify the two allocation rules and demonstrate their properties. Let $\succ$ be some exogenously chosen priority order.

> **Algorithm 1.** ***Naive serial dictatorship (NSD)***. *Inputs:* $(N, H, R, \succ)$. *Let* N *be indexed by their priority order; that is,* $i = 1$ *is the highest priority agent.*

Step 1. *Consider agent* 1*'s favorite objects. Assign one of agent* 1*'s favorite objects to him, and denote it* x_1*. (If there are multiple, the object can be chosen arbitrarily.) Let* $H_1 = H \setminus \{x_1\}$*.*

Step k. *Consider agent* k*'s favorite objects from* H_{k-1}*. Assign one of agent* k*'s favorite objects to him. (If there are multiple, the object can be chosen arbitrarily.) Let* $H_k = H_{k-1} \setminus \{x_k\}$*.*

The dynamic form of NSD is likely familiar to the reader: in order of priority, each agent chooses one desired object. It is immediate that NSD is strategyproof. However, as the following example shows, it is not Pareto efficient nor group strategyproof.

Example 1. Let preferences be as follows, and suppose indifferences are resolved by choosing the lowest indexed object.

R_1	R_2	R_3
1, 2, 3	1	2
	2	1
	3	3

The allocation is $NSD(R) = (1, 2, 3)$. However, it is clear that $x' = (3, 1, 2)$ is feasible and Pareto efficient. It is also achievable by a misreport $3R_1'2R_1'1$, showing NSD is not group strategyproof.

The loss of efficiency (as opposed to the case of strict preferences) is disturbing. NSD is a mis-implementation of the true serial dictatorship mechanism, which asks participants to chose social outcomes (in this case, allocations), with succeeding agents breaking ties. We first note that R_i creates induced preferences R_i^a over allocations X, where

$$xR_i^a x' \iff x_i R_i x_i'$$

We now present true serial dictatorship, which preserves this efficiency.

> **Algorithm 2.** ***True serial dictatorship (TSD)***. *Inputs:* $(N, H, R, \succ)$. *Let* N *be indexed by their priority order; that is,* $i = 1$ *is the highest priority agent.*

Step 1. *Let* $X_1 = \{x \in X : xR_i^a y \ \forall y \in X\}$*. That is,* X_1 *contains all of 1's favorite allocations.*

Step k. *Let* $X_k = \{x \in X_{k-1} : xR_k^a y \ \forall y \in X_{k-1}\}$*. That is,* X_k *contains all of k's favorite allocations remaining from the previous step.*

Step $n+1$*.* *Choose any allocation in* X_n *to implement.*

This is the classic "true" implementation of serial dictatorship over *social outcomes*. A social outcome is an allocation, and agent 1 is implicitly indifferent between all outcomes that give him one of his favorite objects. Agent 1 chooses to preserve his favorite allocations; agent 2 preserves his favorite among the remainder, and so on.

However, this can be difficult to describe for a layperson, since the algorithm acts on allocations. The dynamic form is also unwieldy in practice as the list of allocations can be very long. We present two other implementations of the same mechanism that allows participants to select sets of objects, rather than allocations. The first presents feasible objects via the set of remaining allocations; the second avoids describing allocations at all until the end.

Algorithm 3. ***True serial dictatorship (TSD)***. *Inputs:* $(N, H, R, \succ)$. *Let* N *be indexed by their priority order; that is,* $i = 1$ *is the highest priority agent.*

Step 1. *Consider the set of agent* 1*'s favorite objects from* $H \cup \{\emptyset\}$*, denoted* F_1*. Let* $X_1 = \{x \in X : x_1 \in F_1\}$*.*

Step k*.* *Let* $H_k = \{h \in H \cup \{\emptyset\} : x_k = h \text{ for some } x \in X_{k-1}\}$*. Consider the set of agent* k*'s favorite objects from* H_k*, denoted* F_k*. Let* $X_k = \{x \in X_{k-1} : x_k \in F_k\}$*.*

Step $n + 1$*.* *Choose any allocation in* X_n *to implement.*

Algorithm 3 deals with allocations but presents participants with lists of available objects. At each step t, we check the remaining allocations for objects which can be assigned to agent t. He can select any subset of these and be guaranteed one of them. Participants would only ever select desired objects. Informally, we find it much simpler to explain Algorithm 3 to participants. However, it still requires the *market designer* to deal with allocations. The next algorithm avoids dealing with allocations until the end, in case one needs to be selected arbitrarily.

Algorithm 4. ***True serial dictatorship (TSD)***. *Inputs:* $(N, H, R, \succ)$. *Let* N *be indexed by their priority order; that is,* $i = 1$ *is the highest priority agent.*

Step 1. *Consider the set of agent* 1*'s favorite objects from* $H \cup \{\emptyset\}$*, denoted* F_1*. If* $\emptyset \in |F_1|$*, replace it with a new object labeled* $\emptyset_1$*.*

- *If* $|F_1| = 1$*, then assign* x_1 *as this object. Set* $H_1 = H \setminus \{x_1\}$*. If* $H_1 = \emptyset$*, end the algorithm and assign the remaining agents to* $\emptyset$*.*
- *Otherwise, let* $H_1 = H$*.*

Step k*.* *Consider the set of agent* k*'s favorite objects among* $H_{k-1} \cup \{\emptyset\}$*, denoted* F_k*. If* $\emptyset \in |F_k|$*, replace it with a new object labeled* $\emptyset_k$*.*

- *Let* $K = \{1, ..., k\}$*. For any* $K' \subseteq K$*, if* $|(\cup_{t \in K'} F_t)| = |K'|$*, then agents in* K' *must be assigned to objects in their respective* F_t*, if they were not already.*[a] *Then let* $H_k = H_{k-1} \setminus \{x_t : t \in K'\}$*. If* $H_k = \emptyset$*, end the algorithm and assign agents* $k+1, ..., n$ *to* $\emptyset$*.*
- *Otherwise, let* $H_k = H_{k-1}$*.*

Step $n + 1$*.* *If there are agents who are not yet assigned, choose any allocation compatible with their respective* F_t*.*

[a] This can be found using an algorithm for a maximum bipartite matching algorithm, e.g. Hopcroft–Karp–Karzanov algorithm in $O(k^2)$ time.

This description of serial dictatorship allows participants to claim a set of remaining objects and be guaranteed one of them. As it progresses, it removes sets of objects once they must be assigned. This occurs for

$K' \subseteq K$ if $|\,(\cup_{t\in K'} F_t)\,| = |K'|$, since the number of objects that must be assigned is the same as the number of objects claimed. While it is not explicitly noted in the algorithm, already assigned t can be removed from consideration. Additionally, at step k, only new subsets including k need to be checked, since other subsets were already checked in preceding steps. The following example illustrates Algorithm 4.

Example 2. Let preferences be as follows.

R_1	R_2	R_3	R_4	R_5	R_6
1, 2	1, 2	1	1	1	1
3, 4, 5, 6	3, 4, 5, 6	2	3	2	2
$\emptyset$	$\emptyset$	3, 4, 5	4	3, 5	3, 4, 5
		6	4,5,6	4	$\emptyset, 6$
		$\emptyset$	$\emptyset$	6	

Step 1. Agent 1 chooses among $H \cup \{\emptyset\}$ and submits $F_1 = \{1, 2\}$. Since $|F_1| > 1$, we proceed with $H_1 = H = \{1, 2, 3, 4, 5\}$.

Step 2. Agent 2 chooses among $H_1 \cup \{\emptyset\}$ and submits $F_2 = \{1, 2\}$. Now $|F_1 \cup F_2| = 2$, so we assign agents 1 and 2 to objects in their claims. Let $x_1 = 1, x_2 = 2$.[2] Now $H_2 = H_1 \setminus \{1, 2\} = \{3, 4, 5\}$.

Step 3. Agent 3 chooses among $H_2 \cup \{\emptyset\}$ and submits $F_3 = \{3, 4, 5\}$. $|F_3| = 3 \neq 1$, so proceed with $H_3 = H_2 = \{3, 4, 5\}$.

Step 4. Agent 4 chooses among $H_3 \cup \{\emptyset\}$ and submits $F_4 = \{4\}$. $|F_3 \cup F_4| = 3 \neq 2$, so proceed with $H_4 = H_3 = \{3, 4, 5\}$.

Step 5. Agent 5 chooses among $H_4 \cup \{\emptyset\}$ and submits $F_5 = \{3, 5\}$. $|F_3 \cup F_4 \cup F_5| = 3$. Then we must assign agents 3, 4, and 5. Let $x_3 = 3, x_4 = 4, x_5 = 5$. Now $H_5 = \{6\}$.

Step 6. Agent 6 chooses among $H_5 \cup \{\emptyset\}$ and submits $F_6 = \{\emptyset, 6\}$. We replace it with $F_6 = \{\emptyset_6, 6\}$. $|F_6| = 2 \neq 1$, so we proceed.

Step 7. We choose any compatible allocation for agent 6. We can either assign $x_6 = \emptyset$ or $x_6 = 6$.

The final allocation is $TSD(R) = (1, 2, 3, 4, 5, 6)$. It can be verified that this is a Pareto efficient allocation. However, it is immediate that $NSD(R)$ may be an inefficient allocation. Depending on the tie-breaking rule, agent 3 may be assigned object 4, guaranteeing an inefficient allocation.

We have presented NSD and three descriptions of TSD. It is immediate that Algorithms 2 and 3 are the same, holding fixed the selection in step $n + 1$. To see that Algorithms 3 and 4 are the same, note that Algorithm 3 presents agent k with objects that are compatible with remaining allocations. Algorithm 4 does this analogously by removing objects when they are no longer available to later participants. All three versions have immediate dynamic implementations, where participants are called upon to play sequentially.

[2] $x_1 = 2, x_1 = 1$ also works.

4 Properties of TSD

We now demonstrate the desirable properties of TSD.

Proposition 1. *True serial dictatorship is Pareto efficient and strategyproof.*

Proof. Fix a market $(N, H, R, \succ)$ and let $x = TSD(R)$.

1. **Pareto efficient.** We prove this using the description of Algorithm 2. Suppose there is an allocation y that Pareto dominates x. Consider the highest priority agent i such that $yP_i^a x$. This allocation must not have been remaining in X_{i-1}, otherwise i would have selected it. Thus a higher priority agent k must have removed it, so $xP_k^a y$, a contradiction.

2. **Strategyproof.** We prove this using the description of Algorithm 3. Suppose agent i submits $R_i' \neq R_i$. Denote $TSD(R_i) = x, TSD(R_i', R_{-i}) = x'$. The only change that might affect the allocation is the indifference class reported when i is called upon to play.[3] Denote these IC_i and IC_i'. Note that since $R_{-i} = R_{-i}'$, i faces the same choice. Suppose $IC_i' \setminus IC_i \neq \emptyset$. Then $h \in IC_i' \setminus IC_i$ must be strictly dispreferred (otherwise it was not available or was already in IC_i). Then $x_i' \in IC_i$ or $x_i' \in IC_i' \setminus IC_i$, neither of which is a strict improvement. If $IC_i' \subsetneq IC_i$, then i is still guaranteed to receive one of these objects, so $x_i' I_i x_i$.

$\square$

We also note that TSD is not group strategyproof with the following example.

Example 3. Let R be the following.

R_1	R_2	R_3
1,2	2	1
3	3	2
	1	3

Note that $TSD(R) = (1, 2, 3)$. Let $R_1' = 2R_1 1R_1 3$. Then $TSD(R_1', R_2, R_3) = (2, 3, 1)$. So a group misreport (R_1', R_3) leaves 1 indifferent and 3 strictly better off.

The proof of Proposition 1 illustrates an asymmetry for participants' incentives. While over-reporting acceptable goods is (weakly harmful), it is costless to i to report a subset of his own indifference class. Thus if reporting larger indifference classes requires greater cognitive cost, participants may be tempted to under-report. Informally, a solution to this problem can be to offer payments for larger indifference classes, where payments are of an order of magnitude smaller than benefits of the allocation. In the case of USNA midshipmen, offering, say, \$50 per extra item reported may be enough to induce proper reporting without altering the strategic properties of TSD.

References

Arrow, K.A. 2012. "Social Choice and Individual Values." Yale University Press. New Haven, CT.

Ehlers, L. 2002. "Coalitional Strategy-Proof House Allocation." Journal of Economic Theory 105, 298-317.

Pápai, S. 2003."Strategyproof Assignment by Hierarchical Exchange." Econometrica, 68 (6), 1403-1433.

[3]For clarity, we phrase the procedure as a dynamic implementation, though the idea is the same under either the static or dynamic implementation.