Mechanisms that Play a Game, not Toss a Coin

Toby Walsh
AI Institute, UNSW Sydney
tw@cse.unsw.edu.au

Abstract

Randomized mechanisms can have good normative properties compared to their deterministic counterparts. However, randomized mechanisms are problematic in several ways such as in their verifiability. We propose here to de-randomize such mechanisms by having agents play a game instead of tossing a coin. The game is designed so agents play randomly, and this play injects "randomness" into the mechanism. Surprisingly this de-randomization retains many of the good normative properties of the original randomized mechanism but gives a mechanism that is deterministic and easy, for instance, to audit. We consider three general purpose methods to de-randomize mechanisms, and apply these to six different domains: voting, facility location, task allocation, school choice, peer selection, and resource allocation. We propose a number of novel de-randomized mechanisms for these six domains with good normative properties (such as equilibria in which agents sincerely report preferences over the original problem). In one domain, we additionally show that a new and desirable normative property emerges as a result of de-randomization.

1 Introduction

In 2012, transplant centres across Germany were placed under criminal investigation due to the manipulation of donor waiting lists. Dozens of patients within the Eurotransplant system had preferentially received organs after doctors falsified the severity of their illnesses. The discovery shattered public trust. In the next year, donations dropped by around 30% according to the German organ transplant foundation (DSO) [Walsh, 2018]. Unsurprisingly then, when the author was asked to help update a multi-national mechanism for organ matching, it was made clear that the new mechanism should be simple and trustworthy. Indeed, there was an explicit requirement that it be deterministic to enable auditing [Mattei *et al.*, 2017; Mattei *et al.*, 2018; Walsh, 2022].

Unfortunately deterministic mechanisms can struggle to choose between (perhaps essentially equivalent) alternatives without impacting desirable normative properties like fairness or strategy proofness (e.g. when breaking ties or resolving Condorcet cycles). Randomization offers an escape since we can fairly choose between equivalent alternatives randomly. For example, no deterministic mechanism for house allocation is Pareto efficient, strategy proof and anonymous [Zhou, 1990]. But randomized mechanisms can have all three properties. In particular, the random priority mechanism is (ex post) Pareto efficient, strategy proof and anonymous.

There are, however, many problems with randomization. First, true rather than pseudo randomness is difficult to obtain. We typically require access to some external data source to provide a true source of random bits. Second, if we are making what is an one-off decision then it is inherently difficult to demonstrate that the randomized mechanism was fair. Third, even where decision making is repeated, it can be difficult to audit fairness. We may, for instance, require many executions of the mechanism to have high confidence in its fairness. Fourth, randomization can introduce undesirable computational complexity. For instance, randomness can make it computationally intractable to compute the probability distribution over outcomes (e.g. [Sabán and Sethuraman, 2013]).

We propose an alternative that tackles many of these challenges. Rather than introduce randomness, we stick with deterministic mechanisms but add a game where equilibrium behaviour for agents is to play randomly. We thereby inject randomness into a deterministic mechanism through the agents' play. Surprisingly, we can retain many of the advantages of tossing a coin while avoiding the disadvantages. Indeed, de-randomization can even enhance the normative properties achieved. For instance, we propose a new peer selection mechanism where agents always have an incentive to participate, irrespective of how the other participants vote.

2 Modular Arithmetic Game

At the heart of our general purpose method to de-randomize mechanisms is a simple modular arithmetic game. We first show that equilibrium behaviour in such games is for two (or more) agents to play randomly. The simplest example of such a game is the parity game: two agents play 0 or 1, the even agent wins if the bits are identical, while the odd agent wins if they are different.

Theorem 1. *The parity game has an unique Nash equilibrium in which the even and odd agent both play uniform random bits. The outcome is even/odd with equal probability.*

Proof. No pure strategy is an equilibrium since one agent must lose and therefore can profit by deviating and inverting their strategy from odd to even, or even to odd. To show uniqueness of an uniform random strategy, suppose that the even agent plays odd with probability p and even with probability $1-p$, and that the odd agent gets utility u for a win and v for a loss ($u > v$). To be an equilibrium, the odd agent must get the same reward whether they play odd (expected reward of $(1-p)u + pv$) or even (expected reward of $pu + (1-p)v$) Therefore $(u-v)(1-2p) = 0$. That is, $p = 1/2$. Hence the even agent plays an uniform mixed strategy. A dual argument holds for the odd agent. □

In the more general modular arithmetic game, n agents pick an integer $[0, m)$ where $m \geq n$, and the output is the sum of these integers mod m. First, we show that if one agent plays an uniform mixed strategy, picking an integer uniformly at random, then the output of the game is also an uniform random integer irrespective of the other agents' strategies.

Theorem 2. *If one (or more) agent plays an uniform mixed strategy then the outcome of the modular arithmetic game is an uniform random integer irrespective of how the other agents play.*

Proof. Let $p(j)$ be the probability that agent 1 picks integer j, and $q(j)$ be the probability that agents 2 to m pick integers that sum modular m to j. Note that $\sum_{i=0}^{m-1} p(i) = \sum_{j=0}^{m-1} q(j) = 1$. Suppose that $p(j) = 1/m$. The probability that the output of the game is j is $\sum_{i=0}^{m-1} p(i)q((j-i) \bmod m) = \sum_{i=0}^{m-1} 1/m \; q((j-i) \bmod m) = 1/m \sum_{0}^{m-1} q((j-i) \bmod m) = 1/m \sum_{0}^{m-1} q(i) = 1/m$. □

In modular space, the discrete convolution of any uniform probability distribution with any other probability distribution is the uniform probability distribution. Hence, provided one agent plays an uniform random integer, the output of the modular arithmetic game is itself an uniform random integer.

But are humans any good at playing randomly? Recent empirical studies of rock-paper-scissor games are hopeful, suggesting that humans can be as random as pseudo-random algorithms [Wong *et al.*, 2021]:

> *"... our results demonstrate that human RSG [random sequence generation] can reach levels statistically indistinguishable from computer pseudo-random generators in a competitive-game setting ..."*

We next show that only two agents need to be playing an uniform random integer for play to be in equilibrium. We assume that each agent in the modular arithmetic game has a different most preferred outcome. If each agent has the same preferred outcome, there are trivial pure Nash equilibria (e.g. one agent plays the preferred outcome, and all other agents play 0). On the other hand, if agents have different preferred outcomes, then no pure strategy is an equilibrium. There are, however, mixed Nash equilibria with a simple structure. A **quasi-uniform** strategy is a mixed strategy in which two (or more) agents pick an integer in $[0, m)$ uniformly at random, and other agents play any mixed or pure strategy.

Theorem 3. *Any quasi-uniform strategy is a Nash equilibrium which outputs an uniform random integer in* $[0, m)$.

Proof. Suppose agents play a quasi-uniform strategy. If one agent deviates, then there is still one other agent playing an uniform mixed strategy. The outcome therefore remains an uniform random integer, and the agent's reward is unchanged despite the deviation. Hence any quasi-uniform strategy is an equilibrium. □

There are also mixed Nash equilibria that are not quasi-uniform. Indeed, they are mixed Nash equilibria in which no agent plays an uniform random integer.

Example 1. *Consider $n = m = 4$. Suppose two agents play 0 or 1 with probability $1/2$, and the other two agents play 0 or 2 also with probability $1/2$. Consider one of the agents playing 0 or 1, and another playing 0 or 2. The sum of their plays is an uniform random integer in $[0, 3)$. Hence the sum played by any three of these agents mod 4 is an uniform random integer in $[0, 3)$. Thus any fourth agent has the same reward irrespective of the integer that they play. This is therefore a mixed Nash equilibrium returning an outcome that is an uniform random integer in $[0, 3)$.*

Equilibria like those in the last example require significant coordination between agents. It is likely more realistic to suppose that agents will play an uniform or quasi-uniform strategy as this requires no or limited coordination.

3 Two Simple Examples

We illustrate our idea of de-randomizing mechanisms using a modular arithmetic game with two simple and classic problems in social choice: voting and facility location.

3.1 Random Dictator

We first consider one of the most fundamental problems in social choice, designing mechanisms for voting with good normative problems. As is well known, we quickly run into a wide range of impossibility results. For instance, with three or more candidates, any voting rule that is surjective and strategy proof must also be dictatorial [Gibbard, 1973; Satterthwaite, 1975]. One escape from such impossibility results is randomization. For example, the random dictator mechanism is surjective and strategy proof but the result is not decided by just one voter (i.e. it is not dictatorial). Indeed, it is the only rule that is strategy proof and ex post efficient (i.e. never gives positive probability on Pareto-dominated alternatives) [Gibbard, 1977]. However, voters may not be too keen to see this mechanism being used. The chair might tell a voter: "You preferred candidate lost because some other voter was randomly chosen as dictator", and the voter might have little option but to trust that the chair was fair.

We propose instead the following de-randomization of the random dictator mechanism. Agents submit an integer in $[0, n)$, along with their preferred winner. Let j be the sum of these integers mod n. The dictator is chosen to be the preferred winner of the $j+1$th voter. This de-randomization of the random dictator mechanism is not strategy proof. Any

voter can ensure that they are the dictator by choosing a suitable integer. However, there is a mixed strategy Nash equilibrium in which two (or more) agents choose an integer uniformly at random, as well as all agents report sincerely their most preferred candidate to win. A nice feature of this de-randomization (which we will observe in almost every example explored in this paper) is that an agent's ability to manipulate the outcome is limited to this modular arithmetic game. It is in their best interests to declare their preferences over outcomes (in this case, their preferred winner) sincerely.

3.2 Left-Right-Middle Mechanism

We turn next to the classic problem of facility location, and show how we can return better, higher quality solutions using de-randomization. When locating a single facility on the line, no deterministic and strategy proof mechanism can do better than 2-approximate the maximum cost an agent travels [Procaccia and Tennenholtz, 2013]. However, the randomized left-right-middle (LRM) mechanism 3/2-approximates the maximum cost, and this is optimal as no randomized and strategy proof mechanism can do better. The LRM mechanism selects the leftmost agent with probability 1/4, the midpoint between leftmost and rightmost agents with probability 1/2, and the rightmost agent again with probability 1/4.

We propose the following de-randomization of the LRM mechanism. Agents submit an integer between 0 and 3, along with their location. If the sum of the integers modulo 4 is 0 then the facility is located at the leftmost agent. If the sum of the integers modulo 4 is 1 or 2 then the facility is located at the midpoint between the leftmost and rightmost agents. Otherwise, the sum of the integers modulo 4 is 3 and the facility is located at the rightmost agent.

This de-randomized facility location mechanism is not strategy proof. Suppose, for example, that I am the rightmost agent and I know the other reports. I can ensure the facility is located at my location by submitting a suitable integer. However, there is a mixed strategy Nash equilibrium in which agents choose an integer between 0 and 3 uniformly at random, as well as reporting their sincere location. The expected maximum cost of this mixed strategy is 3/2 times the optimal maximum cost. This is better than that obtained by the best deterministic and strategy proof mechanism.

These two simple examples have illustrated some of the basic ideas in de-randomizing randomized mechanisms. We now apply de-randomization to four other domains where the analysis is more complex. These examples uncover three different but general purpose methods to de-randomize randomized mechanisms. In the first ("game-first"), we play a modular arithmetic game to pick a random "seed". This is then applied to the original randomized mechanism. In the second ("game-last)", we apply a randomized mechanism to generate a probabilistic outcome. We then play a modular arithmetic game to convert this into a discrete ex post outcome. And in the third ("game-interleaved"), we interleave playing a modular arithmetic game with applying the mechanism.

4 Task Allocation

The first more complex domain that we consider is task allocation. There are m tasks that need to be allocated to 2 agents. Agent i declares that task j will take time t_j^i. The goal is to allocate tasks to agents to minimize the makespan (the completion time of the last task to either agent). To compensate agents for performing a task, agents receive a payment. The payment is in hours of work, and the overall utility of an agent is this payment less the actual amount of work performed. The mechanism design problem here is to devise a mechanism which approximates well the optimal makespan and incentivizes agents to report the time it will take them to execute each task sincerely.

Algorithm 1 BiasedMinWork(m, t_j^i, b_j)

```
a_1, a_2 := {}                                  ; task allocation
p_1, p_2 := 0                                   ; payment
for j = 1 to m do
    i := 1+b_j, i' := 3−i                       ; construct permutation
    if t_j^i ≤ (4/3) t_j^{i'} then
        a_i := a_i ∪ {j}, p_i := p_i + (4/3) t_j^{i'}
    else
        a_{i'} := a_{i'} ∪ {j}, p_{i'} := p_{i'} + (3/4) t_j^i
    end if
end for
return a_1, a_2, p_1, p_2
```

Nisan and Ronen [2001] prove that no deterministic mechanism can better than 2-approximate the makespan, and that a simple VGC-style min-work mechanism that allocates each task to the quickest agent, paying them the time that would be taken by the other agent to perform the task is strategy proof and achieves this 2-approximation. They also prove that randomization can improve this approximation ratio in expectation. In particular, they show that the biased min-work mechanism (Algorithm 1) provides a 7/4-approximation of the optimal makespan in expectation when given m random bits drawn uniformly. This mechanism is strongly truthful (i.e. even if we know the random bits, it remains strategy proof). We denote these random bits by b_j.

To de-randomize this biased min-work mechanism, we have agents set the bits b_j by means of a simple parity game. In particular, we suppose the mechanism operates in m rounds. In the jth round, both agents submit their time to complete task j. We additionally suppose they now submit a single bit, 0 or 1. We set b_j to be the xor of these two bits. This example thus illustrates the "game-interleaved" method to de-randomize a mechanism.

Theorem 4. *This de-randomized biased min-work mechanism has an unique mixed subgame perfect Nash equilibrium in which agents submit bits uniformly at random, and sincerely report their task times. This $\frac{7}{4}$-approximates the optimal makespan in expectation.*

Proof. Consider agent 1 in round j. We say that agent 1 wins the round iff $b_j = 0$. There are two cases. In the first, $t_j^1 \leq \frac{4}{3} t_j^2$. If agent 1 wins this round (that is, $b_j = 0$), then agent 1 pays a cost of t_j^1 but receives a greater or equal payment of $\frac{4}{3} t_j^2$. Agent 2, on the other hand, loses this round, incurs no cost but receives no payment. Suppose agent 2 in-

stead wins the round (that is, $b_j = 1$). There are two subcases. In the first subcase, $\frac{3}{4}t_j^1 \leq t_j^2 \leq \frac{4}{3}t_j^1$. Agent 1 is now not allocated the task and agent 1 therefore has no additional cost or payment. However, agent 2 is allocated the task. Agent 2 receives a payment of $\frac{4}{3}t_j^1$ which is greater than their cost. Hence, in the first subcase, both agent 1 and 2 want to win the round. In the second subcase, $t_j^2 > \frac{4}{3}t_j^1$ and agent 1 is allocated task j. Their payment is $\frac{3}{4}t_j^2$. This is strictly greater than their cost, t_j^i. However, it is a smaller payment than when agent 1 wins the round. Hence, it was desirable for both agents to have won this round. The other case, in which $t_j^1 > \frac{4}{3}t_j^2$ is dual. The mechanism treats agents identically so agent 2 also wants to win each round. It follows that the unique mixed subgame perfect Nash equilibrium has both agents submitting bits uniformly at random. The biased minwork mechanism is strongly truthful as, even if the result of the parity game is known, agents have no incentive to misreport their task time. As the subgame perfect Nash equilibrium has agents winning each round with equal probability, the mechanism returns the same distribution of task allocations as the randomized mechanism. Hence, it $\frac{7}{4}$-approximates the optimal makespan in expectation. □

5 Peer Selection

We consider next the peer selection problem in which agents have to choose one amongst themselves to receive a prize (e.g. [Alon *et al.*, 2011; Merrifield and Saari, 2009; Aziz *et al.*, 2016; Aziz *et al.*, 2019; Walsh, 2014]). For example, a committee might want to choose one person within the committee to be chair. As a second example, a school class might want to choose a class representative. As a third example, the members of an academic society might want to choose a member to receive a medal. We propose here a new peer selection mechanism with a novel normative property encouraging participation: each one of the agents is guaranteed to have a say in the winner. That is, irrespective of how the other agents vote, every agent can change the winner of the prize. This peer selection mechanism is based on the idea of sequentially eliminating agents until just one is left who wins the prize. Our results easily generalize to the more general voting setting where n agents vote over m candidates.

We first consider a deterministic peer selection mechanism in which agents successively eliminate candidates. This mechanism lacks the desirable normative property of anonymity (i.e. permuting the names of the agents changes the outcome). We therefore randomize it to give anonymity. We will then de-randomized this mechanism to give a new deterministic and anonymous mechanism. This de-randomized mechanism has the property that, even if some agent has a large majority support, a dissenting agent still has an incentive to participate and change the outcome. This example is an instance of the "game-first" method for de-randomizing a randomized mechanism in which we play a modular arithmetic game first to select a random "seed".

5.1 Sequential Elimination

We start by considering the strategy behaviour of agents with the simple deterministic sequential elimination (SE) mechanism studied by Moulin [1983] and others. We will need this equilibrium result to discuss the equilibria of the de-randomized mechanism that we will shortly introduce. The SE mechanism starts with all agents as possible winners, and then has the agents in a given order eliminate one of the remaining agents from the set of possible winners until only a single agent remains. We suppose the n agents have strict preferences so there is, for example, always an unique least preferred agent to eliminate. This SE mechanism is in some sense a dual of the dictatorship mechanism. Whilst the dictatorship mechanism is strategy proof, sequential elimination is not. A agent may not eliminate their worst ranked agent if a later agent will. Strategic play is, however, easily computed.

Given a fixed elimination ordering, we can view the SE mechanism as a repeated game. Supposing agents have complete information about the elimination ordering and other preferences, the subgame perfect Nash equilibrium of this game can be computed using backward induction. However, such a computation is exponential in m. There is, however, a simple linear time method to compute the subgame perfect Nash equilibrium. We simply reverse the elimination ordering and play the game backwards [Moulin, 1983]. This is similar to equilibria of the sequential allocation mechanism itself [Kalinowski *et al.*, 2013; Walsh, 2016; Aziz *et al.*, 2017].

It is easy to see informally why this is the case. Agents can play strategically to ensure that the last agent to be eliminated is the least preferred agent of the last agent in the elimination ordering. The last agent will surely want this agent to be eliminated. Therefore earlier agents in the elimination order might strategically not eliminate this agent, even if it is also their least preferred agent. An early agent will profit by eliminating some more preferred agent, safe in the knowledge that their least preferred agent will be eliminated at the end of the game. Similarly, if we discount this agent, the agents can play strategically to ensure that the penultimate agent to be eliminated is the least preferred agent of the penultimate agent in the elimination ordering, and so on. Indeed, all subgame perfect Nash equilibria return the same winner, the one computed by this reversed computation.

5.2 Random Sequential Elimination

The SE mechanism that we just considered is not anonymous. It treats agents differently according to where they appear in the elimination ordering. We can make it anonymous by randomizing over the agents. In particular, we propose the random sequential elimination (RSE) mechanism which picks a random order of agents and then has agents eliminate agents one by one using this ordering until just one agent remains who is declared the winner.

We can de-randomize RSE using a "game-first" method. In the first stage, the n agents play a modular arithmetic game in which each submits an integer in $[0, n!)$. We use the result of this game to pick one of the $n!$ priority orderings of agents via an inverse Lehmer code. In the second stage, we eliminate agents using this ordering. With this de-randomized RSE mechanism, agents have only limited options for strategic play. In particular, the voting game for the de-randomized mechanism has a mixed subgame perfect Nash equilibrium in

which agents pick random integers uniformly, and then eliminate their least preferred remaining agent in reverse order.

Theorem 5. *The voting game of the de-randomized RSE mechanism has a mixed subgame perfect Nash equilibrium in which two or more agents pick numbers uniformly at random in the first stage, and then agents sincerely eliminate their least preferred agent remaining in the reverse elimination order in the second stage.*

A novel property of this mechanism is that agents always have an incentive to participate irrespective of the votes of the other agents. We say that a peer selection mechanism over two or more agents is **responsive** iff, regardless of the votes of the other agents, there exist two different ways that any agent can vote with different outcomes. Note that the SE mechanism is not responsive as the last agent in the elimination order cannot change the outcome. The de-randomized RSE mechanism is, on the other hand, responsive.

Theorem 6. *The de-randomized RSE mechanism is responsive.*

Proof. Consider the first elimination round and any agent. There are two cases. In the first case, the agent is chosen to perform the first elimination. By changing the agent that is eliminated to be the current winner, the overall winner must change. In the second case, the agent is not chosen to perform the first elimination. Suppose this agent submits a different integer to ensure that they perform the first elimination. If the agent now eliminates the current winner, then the overall winner must again change. □

Another desirable normative property in peer selection is impartiality [Holzman and Moulin, 2013]. A peer selection mechanism is **impartial** iff an agent cannot influence whether they win or not. For example, we can design an impartial mechanism by partitioning agents into two sets, having each set vote on the other, and then randomly choosing between the two possible winners. We can de-randomize this mechanism without violating impartiality by having the agents who are not one of the two possible winners playing a parity game to pick between the two possible winners. The resulting "game-last" mechanism is impartial but is not responsive. On the other hand, the de-randomized RSE mechanism is responsive but not impartial. This is to be expected as impartiality and responsiveness are impossible to achieve simultaneously.

Theorem 7. *No deterministic peer selection mechanism is both responsive and impartial.*

Proof. With two agents, the only impartial mechanism selects a fixed winner. This is not responsive. Consider then three or more agents, and any reports for these agents. Pick any agent. Suppose this agent wins. If the mechanism is responsive, there must be some other report for this agent that changes the winner. Consider the agent changing from this new report to its original report. This change violates impartiality. This is a contradiction. Hence, the assumption that the agent wins is false. But this is true for every possible agent. Thus, no agent can win. □

We end this section with some important related work. Bouveret *et al.* [2017] previously studied the SE mechanism for the more general voting setting. They argue that the SE mechanism has low-communication complexity and, with a good elimination ordering, can have good properties such as returning the Borda winner. Here we considered such a mechanism where the elimination ordering is chosen randomly. The resulting randomized mechanism is now anonymous (as is the de-randomized version).

6 School Choice

We turn next to a very practical social choice problem that impacts citizens of many different countries. In school choice, we consider the two sided matching problem in which schools and students choose each other (e.g. [Abdulkadiroğlu and Sönmez, 2003; Ergin and Sonmez, 2006; Abdulkadiroglu *et al.*, 2005]). One of the most popular mechanisms for school choice in every day use around the world is the student-proposing deferred acceptance (DA) mechanism of [Gale and Shapley, 1962]. This has many desirable normative properties such as strategy proofness for the students (but not schools) and stability of the solution.

One issue with the DA mechanism is that it supposes schools have a complete ordering over students. In practice, schools often only have broad preferences over students. For example, those with siblings at the school might be strictly preferred to those merely within the school district, and these two groups might be strictly preferred to those outside the school district. However, within each group, schools might be indifferent between students. It is therefore common to order students within each group using a random lottery.

To de-randomize the DA mechanism which uses such a random lottery, we could have the n students in a particular group instead play a modular arithmetic game by submitting an integer in $[0, n!)$. We then use the result of this game to pick one of the $n!$ priority orderings of students in this group via an inverse Lehmer code. This school choice example thus illustrates the game-first method to de-randomize a randomized mechanism: we first play a modular arithmetic game to construct a random "seed" (ordering) which is then used in the second step by the original mechanism.

While this game is polynomial, requiring students to submit just $O(n \log n)$ bits, it may nevertheless be prohibitively expensive. For instance, with 1000 students to order in a group, we would require a student to submit an integer with several thousand decimal digits (as $1000! \approx 4.0 \text{ x } 10^{2568}$). We propose a more efficient mechanism where, instead of each student submitting $O(n \log n)$ bits, each of the n students submits only $O(\log n)$ bits. These are then combined to form a (random) $O(n \log n)$ priority order. The mechanism differs on whether n is odd or even. We suppose students are numbered from 0 to $n-1$.

If $n = 2k$, student 0 submits an integer $b_0 \in [0, n)$, each student $i \in [1, n-1)$ submits two integers, $a_i \in [0, i]$ and $b_i \in [0, n-i]$, and the final student $n-1$ submits an integer $a_{n-1} \in [0, n)$. We construct a permutation ordering as follows. The first student in the ordering is $(a_{n-1} + b_0) \bmod n$. The second student is then the $(a_{n-2}+b_1) \bmod (n-1)$ largest

remaining student (counting from zero). The third student in the ordering is then the $(a_{n-3} + b_2)$ mod $(n-2)$ largest remaining student, and so on.

If $n = 2k+1$, student 0 submits two integers $a_0 \in [0,k]$ and $b_0 \in [0,n)$, each student $i \in [1, n-1)$ submits two integers, $a_i \in [0,i]$ and $b_i \in [0, n-i]$, and the final student $n-1$ submits two integers $a_{n-1} \in [0,n)$ and $b_{n-1} \in [0,k]$. We construct a permutation from these as follows: the first student in the permutation is $(a_{n-1} + b_0)$ mod n. The second student is then the $(a_{n-2} + b_1)$ mod $(n-1)$ largest remaining student (counting from zero), and so on. There is, however, one exception in the exact middle of the permutation as the two integers being added together would otherwise be submitted by the same agent. More precisely, the $k+1$th student in the permutation is computed not as the $(a_k + b_k)$ mod $(k+1)$ largest remaining student but instead as the $(a_k + b_k + a_0 + b_{n-1})$ mod $(k+1)$ largest.

In the second stage of our de-randomized mechanism, we run the regular DA mechanism using this priority order to break ties that a school has within a group. This de-randomized mechanism inherits the stability of the underlying DA mechanism. In addition, whilst students can act strategically, their strategic behaviours are limited to how they play the modular arithmetic game.

Theorem 8. *The de-randomized DA mechanism has a mixed Nash equilibrium in which two or more students in the first stage select integers with uniform probability, and then all students select schools in the second stage sincerely. This equilibrium corresponds to a probability distribution over ex post stable matchings.*

7 Resource Allocation

Our final application domain is resource allocation where we consider (randomized) mechanisms for the allocation of indivisible items [Walsh, 2020]. We have n agents who have preferences over m indivisible items. Our goal is to allocate items whole to agents according to these preferences. In this setting, randomization allows us to deal with contested items fairly. For example, rather than unilaterally give a contested item to one agent, we could toss a coin to decide which agent gets it. Two of the most prominent randomized mechanisms with good normative properties for this domain are the probabilistic serial (PS) and random priority (RP) mechanisms.

7.1 Probabilistic Serial

In the probabilistic serial (PS) mechanism agents simultaneously "eat" at a constant speed their most preferred remaining item [Bogomolnaia and Moulin, 2001]. This gives a randomized or probabilistic assignment which can easily be realized as a probability distribution over discrete allocations. Unfortunately, the PS mechanism is not strategy proof (see [Aziz *et al.*, 2015b; Aziz *et al.*, 2015a]). However, it has good welfare and efficiency properties. It is, for instance, SD-efficient and SD-envyfree[1].

To de-randomize the PS mechanism, we first identify how much precision is needed to represent the probabilities in the random assignment it generates. This precision will dictate the size of the modular aritmetic game that agents will play. We prove here that the PS mechanism only needs a polynomial number of bits with which to represent probabilities.

Theorem 9. *For every one of the n agents and of the m items, there exists an integer k such that the PS mechanism allocates the item to the agent with probability $\frac{k}{(n!)^m}$.*

Proof. We suppose without loss of generality that items are fully "consumed" by the PS mechanism in numerical order. We can consider then m steps of the PS mechanism, each ending with a new item being fully consumed. We assume that only one item is ever fully consumed at a time. We will discuss relaxing this assumption shortly.

Let $k_{i,j}$ be the number of agents eating the ith item at the jth step of the probabilistic serial mechanism. For notational simplicity, we write k_i for $k_{i,i}$. Hence, k_i is the number of agents eating the ith item when it is fully consumed. Note that $k_i \in [1,n]$ and $k_{i,j} \in [0,n)$ for $i \neq j$.

The first step takes $\frac{1}{k_1}$ time for the first item to be fully consumed. The k_1 agents each get a share of $\frac{1}{k_1}$ of probability of this first fully consumed item. Note that this is an integer multiple of $\frac{1}{n!}$, and thus also of $\frac{1}{(n!)^m}$. Consider any item $j > 1$. The $k_{j,1}$ agents eating this item each get a share of $\frac{1}{k_1}$ of probability of item j. Note that this is again an integer multiple $\frac{1}{(n!)^m}$. There is now $1 - \frac{k_{j,1}}{k_1}$ of item j left. That is, $\frac{k_1 - k_{j,1}}{k_1}$ of item j left. Note this is an integer multiple of $\frac{1}{n!}$

Supposing $m \geq 2$, during the next step, k_2 agents eat whatever remains of the second item. This takes $(\frac{k_1 - k_{2,1}}{k_1})/k_2$ time. Each of the k_2 agents eating this item thereby receives a share of $\frac{k_1 - k_{2,1}}{k_1 k_2}$ of probability of item 2. Note this is an integer multiple of $\frac{1}{(n!)^2}$, and thus of $\frac{1}{(n!)^m}$. Consider any other item $j > 2$. The $k_{j,1}$ agents eating this item each get a share of $\frac{k_1 - k_{2,1}}{k_1 k_2}$ of probability of item j. Note that this is again an integer multiple $\frac{1}{(n!)^m}$. There is now $\frac{k_1 - k_{j,1}}{k_1} - \frac{k_{j,1}(k_1 - k_{2,1})}{k_1 k_2}$ of item j left. That is, $\frac{k_1 k_2 - k_{j,1}(k_1 + k_2) + k_{j,1} k_{2,1}}{k_1 k_2}$. Note that this is an integer multiple of $\frac{1}{(n!)^2}$.

The argument repeats in subsequent steps. In the jth step, agents add an additional probability for an item which is an integer multiple of $\frac{1}{(n!)^j}$. And the amount left of any item not fully consumed is also an integer multiple of $\frac{1}{(n!)^j}$. Adding together all the probabilities, we conclude at the final mth step that for each agent and item, there exists an integer k such that the PS mechanism allocates the item to the agent with probability $\frac{k}{(n!)^m}$. Note that if two or more items are

[1]These are efficiency and fairness notions that are defined for ordinal preferences. In particular, the *SD* (stochastic dominance) ordering prefers an allocation p to an agent over q if the probability for the agent to get the top i items in p is at least as large as in q for all $i \in [1,m]$. If an allocation to an agent is *SD-preferred* over another then it has greater or equal expected utility for all utilities consistent with the agent's ordinal preferences. Notions like *SD-efficiency*, *SD-envy freeness* and *SD-strategy proofness* can be defined in the natural way from the SD-preference ordering.

fully consumed at exactly the same time, the argument is similar. However, we now have strictly less than m steps but the conclusion still holds. Note that $(n!)^m$ can be represented in $O(mn\log(n))$ bits which is polynomial in n and m. □

We next define a two stage mechanism that de-randomizes the PS mechanism. In the first stage, the mechanism constructs the probabilistic allocation of the usual PS mechanism. In the second stage, agents play a modular arithmetic game, and the mechanism uses the outcome of this game to construct a discrete ex post outcome. This illustrates the "game-last" method to de-randomize a randomized mechanism.

In more detail, in the second stage, each agent submits an integer in $[0,(n!)^m)$. Let σ be the sum of these integers mod $(n!)^m$. The mechanism then allocates item j to agent $\min\{k \mid \sum_{i=1}^{k} p_{i,j} \geq \frac{\sigma}{(n!)^m}\}$ where $p_{i,j}$ is the probability that agent i is allocated item j in the probabilistic allocation. In other words, the mechanism treats $\sigma/(n!)^m$ as a random draw. Note that this is a worst case. As this is a "game-last" de-randomization, we can analyse the probabilistic allocation returned in the first stage and perhaps use a smaller range of integers than $[0,(n!)^m)$. The second stage has a mixed Nash equilibrium in which agents choose an integer uniformly at random. This gives each agent an expected welfare equal to that of the randomized allocation computed in the first stage.

Theorem 10. *A mixed Nash equilibrium of the second stage has two or more agents select integers in $[0,(n!)^m)$ with uniform probability.*

As with the PS mechanism itself, agents may strategically misreport their ordinal preferences over items in the first stage of this de-randomized PS mechanism. For example if one agent prefers item 1 to 2 and 2 to 3, and a second agent prefers 2 to 3 and 3 to 1, then the first agent can profitably misreport that they prefer 2 to 1 and 1 to 3. A pure Nash equilibrium of the PS mechanism is guaranteed to exist, but is NP-hard to compute in general [Aziz *et al.*, 2015a]. Indeed, even computing a best response is NP-hard [Aziz *et al.*, 2015b].

One tractable special case is two agents since there exists a linear time algorithm to compute a pure Nash equilibrium with two agents which yields the same probabilistic allocation as the truthful profile [Aziz *et al.*, 2015a]. As the PS mechanism is envy-free ex ante, it follows that a mixed Nash equilibrium with two agents consisting of this pure strategy for the ordinal preferences and an uniform mixed strategy for the modular arithmetic game is envy-free ex ante. Another tractable special case is identical ordinal preferences. The PS mechanism is strategy proof in this case. Hence, a combined mixed Nash equilibrium with identical ordinal preferences (consisting of a sincere strategy for the ordinal preferences and an uniform mixed strategy for the modular arithmetic game) is again envy-free ex ante.

7.2 Random Priority

For simplicity, we consider house allocation (i.e. one item to allocate per agent). However, our results easily generalize to a setting with more (or fewer) items than agents. The random priority (RP) mechanism picks a random order of agents, and then agents take turns according to this order to pick their most preferred remaining item (house) (e.g. [Abdulkadiroglu and Sonmez, 1998; Zhou, 1990]).

Random priority is used in many real world settings. For example, the author was allocated undergraduate housing using the RP mechanism. RP is one of the few strategy proof mechanisms available for house allocation. Indeed, any strategy proof, nonbossy[2] and neutral[3] mechanism is necessarily some form of serial dictatorship like the RP mechanism [Svensson, 1999].

We define a two stage "game-first" mechanism that de-randomizes the RP mechanism. In the first stage, agents play a modular arithmetic game, and the mechanism uses the outcome of this game to construct a picking ordering for the second stage. Each agent submits an integer in $[0,n!)$. We sum these integers mod $n!$ and then convert this into a permutation ordering of agents via an inverse Lehmer code. In the second stage, the mechanism uses this ordering with a serial dictatorship mechanism to allocate items to agents. This is then another example of a "game-first" method to de-randomize a randomized mechanism.

Theorem 11. *The de-randomized RP mechanism has a mixed Nash equilibrium in which two or more agents in the first stage select integers in $[0,n!)$ with uniform probability, and all agents then select items in the second stage sincerely.*

Like RP which returns a probability distribution over ex post outcomes that are Pareto efficient, the de-randomized RP mechanism is Pareto efficient. Note that we could play the modular arithmetic game second, after we first compute the randomized allocation returned by the RP mechanism, and then use the outcome of the modular arithmetic game to perform a "random draw" like we did when de-randomizing the PS mechanism. However, it is NP-hard to compute the probabilistic allocation returned by RP [Saban and Sethuraman, 2015]. Therefore this is not a tractable de-randomization.

8 Conclusions

We have proposed three related methods of de-randomizing mechanisms: "game-first", "game-last" and "game-interleaved". Each introduces a modular arithmetic game which can inject randomness into the mechanism via the randomness in agents' play. Surprisingly, these de-randomized mechanisms retain many of the good normative properties of the underlying randomized mechanism. For instance, all but one of the de-randomized mechanisms is "quasi-strategy proof" as they have mixed Nash equilibria in which agents play the game randomly but report other preferences sincerely. We demonstrate how these de-randomization methods work in six different domains: voting, facility location, task allocation, school choice, peer selection, and resource allocation. In one case, de-randomization additionally introduced a new and desirable normative property (namely that the de-randomized peer selection mechanism was responsive always to the preferences of any agent).

[2] A mechanism is *nonbossy* if an agent cannot change the allocation without changing their own allocation.

[3] A mechanism is *neutral* if permuting the names of the items merely permutes the outcome

Acknowledgements

The author is supported by the Australian Research Council through an ARC Laureate Fellowship FL200100204.

References

[Abdulkadiroglu and Sonmez, 1998] A. Abdulkadiroglu and T. Sonmez. Random serial dictatorship and the core from random endowments in house allocation problems. *Econometrica*, 66(3):689–702, May 1998.

[Abdulkadiroglu *et al.*, 2005] A. Abdulkadiroglu, P.A. Pathak, A.E. Roth, and T. Sonmez. The Boston Public School Match. *American Economic Review*, 95(2):368–371, 2005.

[Abdulkadiroğlu and Sönmez, 2003] A. Abdulkadiroğlu and T. Sönmez. School choice: A mechanism design approach. *American Economic Review*, 93(3):729–747, 2003.

[Alon *et al.*, 2011] N. Alon, F. Fischer, A. Procaccia, and M. Tennenholtz. Sum of us: Strategyproof selection from the selectors. In *Proceedings of the 13th Conferene on Theoretical Aspects of Rationality and Knowledge*, pages 101–110, 2011.

[Aziz *et al.*, 2015a] H. Aziz, S. Gaspers, S. Mackenzie, N. Mattei, N. Narodytska, and T. Walsh. Equilibria under the probabilistic serial rule. In Q. Yang and M. Wooldridge, editors, *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, (IJCAI 2015)*, pages 1105–1112, 2015.

[Aziz *et al.*, 2015b] H. Aziz, S. Gaspers, S. Mackenzie, N. Mattei, N. Narodytska, and T. Walsh. Manipulating the probabilistic serial rule. In G. Weiss, P. Yolum, R.H. Bordini, and E. Elkind, editors, *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015)*, pages 1451–1459. ACM, 2015.

[Aziz *et al.*, 2016] Haris Aziz, Omer Lev, Nicholas Mattei, Jeffrey S. Rosenschein, and Toby Walsh. Strategyproof peer selection: Mechanisms, analyses, and experiments. In Dale Schuurmans and Michael P. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 397–403. AAAI Press, 2016.

[Aziz *et al.*, 2017] Haris Aziz, Paul Goldberg, and Toby Walsh. Equilibria in sequential allocation. In Jörg Rothe, editor, *Proceedings of 5th International Conference on Algorithmic Decision Theory, ADT 2017*, volume 10576 of *Lecture Notes in Computer Science*, pages 270–283. Springer, 2017.

[Aziz *et al.*, 2019] Haris Aziz, Omer Lev, Nicholas Mattei, Jeffrey S. Rosenschein, and Toby Walsh. Strategyproof peer selection using randomization, partitioning, and apportionment. *Artif. Intell.*, 275:295–309, 2019.

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

[Bouveret *et al.*, 2017] Sylvain Bouveret, Yann Chevaleyre, François Durand, and Jérôme Lang. Voting by sequential elimination with few voters. In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 128–134. ijcai.org, 2017.

[Ergin and Sonmez, 2006] H. Ergin and T. Sonmez. Games of school choice under the Boston mechanism. *Journal of Public Economics*, 90(1-2):215–237, January 2006.

[Gale and Shapley, 1962] D. Gale and L.S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69:9–15, 1962.

[Gibbard, 1973] A. Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41:587–601, 1973.

[Gibbard, 1977] Allan Gibbard. Manipulation of Schemes That Mix Voting with Chance. *Econometrica*, 45(3):665–681, April 1977.

[Holzman and Moulin, 2013] R. Holzman and H. Moulin. Impartial nominations for a prize. *Econometrica*, 81(1):173–196, 2013.

[Kalinowski *et al.*, 2013] T. Kalinowski, N. Narodytska, T. Walsh, and L. Xia. Strategic behavior when allocating indivisible goods sequentially. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence (AAAI 2013)*. AAAI Press, 2013.

[Mattei *et al.*, 2017] N. Mattei, A. Saffidine, and T. Walsh. Mechanisms for online organ matching. In *Proceedings of the 26th International Joint Conference on AI*. International Joint Conference on Artificial Intelligence, 2017.

[Mattei *et al.*, 2018] N. Mattei, A. Saffidine, and T. Walsh. Fairness in deceased organ matching. In *Proceedings of 1st AAAI/ACM Conference on AI, Ethics, and Society*, 2018.

[Merrifield and Saari, 2009] M.R. Merrifield and D.G. Saari. Telescope time without tears: a distributed approach to peer review. *Astronomy & Geophysics*, 50(4):4.16–4.20, 2009.

[Moulin, 1983] Hervé Moulin. *Advanced Textbooks in Economics, Volume 18: The strategy of social choice*. North Holland, 1983.

[Nisan and Ronen, 2001] Noam Nisan and Amir Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35(1):166–196, 2001.

[Procaccia and Tennenholtz, 2013] A.D. Procaccia and M. Tennenholtz. Approximate mechanism design without money. *ACM Trans. Econ. Comput.*, 1(4):18:1–18:26, December 2013.

[Sabán and Sethuraman, 2013] D. Sabán and J. Sethuraman. The complexity of computing the random priority allocation matrix. In Y. Chen and N. Immorlica, editors, *Web and Internet Economics - 9th International Conference, WINE 2013, Cambridge, MA, USA, December 11-14, 2013, Proceedings*, volume 8289 of *Lecture Notes in Computer Science*, page 421. Springer, 2013.

[Saban and Sethuraman, 2015] Daniela Saban and Jay Sethuraman. The complexity of computing the random priority allocation matrix. *Mathematics of Operations Research*, 40(4):1005–1014, 2015.

[Satterthwaite, 1975] M. Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10:187–216, 1975.

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

[Walsh, 2014] T. Walsh. The PeerRank Method for peer assessment. In Torsten Schaub, editor, *Proc. of the 21st European Conference on Artificial Intelligence (ECAI-2012)*, Frontiers in Artificial Intelligence and Applications. IOS Press, 2014.

[Walsh, 2016] T. Walsh. Strategic behaviour when allocating indivisible goods. In D. Schuurmans and M. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 4177–4183. AAAI Press, 2016.

[Walsh, 2018] Toby Walsh. *2062: The World that AI Made*. La Trobe University Press, 2018.

[Walsh, 2020] Toby Walsh. Fair division: The computer scientist's perspective. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4966–4972. International Joint Conferences on Artificial Intelligence Organization, 7 2020.

[Walsh, 2022] Toby Walsh. *Machines Behaving Badly: The Morality of AI*. La Trobe University Press, 2022.

[Wong *et al.*, 2021] Alice Wong, Garance Merholz, and Uri Maoz. Characterizing human random-sequence generation in competitive and non-competitive environments using Lempel–Ziv complexity. *Scientific Reports*, 11:Art. No. 20662, Oct 2021. Funding by John Templeton Foundation.

[Zhou, 1990] L. Zhou. On a conjecture by Gale about one-sided matching problems. *Journal of Economic Theory*, 52(1):123 – 135, 1990.