Probabilistic Boston Mechanism

Xiang Han* Shanhui Yang*

October 2023

Abstract

The popular Boston mechanism in school choice satisfies two main desirable axioms, Pareto efficiency and respect for (preference) rank. However, when priorities are weak, the Boston mechanism with random tie-breaking no longer has these properties ex-ante. In the context where all agents have equal priorities and preferences are weak, we propose a new random mechanism, the probabilistic Boston mechanism, to restore the two desiderata. The mechanism combines features from both the Boston mechanism and the probabilistic serial rule, and satisfies stochastic-dominance efficiency, (ex-ante) respect for rank, as well as a new fairness notion of equal-rank ordinal fairness that is stronger than equal treatment of equals.

Keywords: Boston mechanism; random allocation; weak preferences; respect for rank; stochastic-dominance efficiency; probabilistic serial rule

JEL Codes: C78; D47; D61; D63

1 Introduction

The *Boston mechanism* is still one of the most common and popular centralized mechanisms in placing students to schools around the world, although the school choice literature starting from Abdulkadiroğlu and Sönmez (2003) has largely supported the use of alternative and superior solutions.[1] In particular, the *deferred acceptance algorithm* (DA)

*Both Han and Yang are affiliated with the School of Economics, Shanghai University of Finance and Economics. Emails: han.xiang@sufe.edu.cn (Han), shanhuiyangsufe@163.com (Yang).

[1]For instance, the Boston mechanism is still used in Belgium, Germany, Spain, and Cambridge and Seattle in the US (Terrier et al., 2021). Moreover, a hybrid of the Boston mechanism and the deferred acceptance algorithm is adopted in Chinese college admissions (Chen and Kesten, 2017).

from Gale and Shapley (1962), which has replaced it in many regions, is strategy-proof and agent-optimal stable, while the Boston mechanism is manipulable and unstable. Such drawbacks of the Boston mechanism are attributed to the *immediate-acceptance* nature in its allocation procedure: at each step k a school only admits students who list it as the kth choice, and it does not reject any previously accepted student even if there are currently applicants with higher priorities.

However, such procedure also indicates that the outcome is Pareto efficient, while DA is only constrained efficient. Moreover, as revealed by earlier axiomatic characterizations (Kojima and Ünver, 2014, Doğan and Klaus, 2018), a fundamental feature of the procedure is that it *respects preference ranks*, i.e., if a student envies the assignment of another student, then the latter must rank her assigned school weakly higher in her preference list. Therefore, from an axiomatic perspective, the popularity of the Boston mechanism mainly comes from this welfare criterion, in addition to Pareto efficiency.[2]

In the presence of ties in priorities, which are often observed in real-life school choice problems, random allocations are needed due to fairness considerations regarding equal claims. Then a natural and common solution is to pick an ordering of the agents from the uniform distribution to break the ties in priorities, and then apply the Boston mechanism. Although such randomized Boston mechanism satisfies the minimum fairness requirement of equal treatment of equals, as shown in Chen et al. (2023), it does not have the two attractive properties of the original Boston mechanism ex-ante: it is not stochastic-dominance (sd) efficient, and does not satisfy *respect for rank*, which requires that if student i is admitted to school a with positive probability and student j is admitted to a worse school with positive probability, i.e., j envies the probability share of a received by i, then i must rank a weakly higher than j.[3]

In this study, we propose the *probabilistic Boston mechanism* (PB) that restores the above two desiderata, in the simplified *house allocation* setup (Hylland and Zeckhauser, 1979), where all agents (students) have equal priority for each indivisible object (school). In addition, we allow indifferences in preferences. As far as we know, even the original deterministic Boston mechanism has not been extended to the general weak preference

[2]There is also abundant empirical evidence which suggests that the overall welfare can be higher under the Boston mechanism than under DA. See, for example, He (2017), Agarwal and Somaini (2018) and Calsamiglia et al. (2020).

[3]More generally, randomizations over deterministic mechanisms often fail to satisfy desirable ex-ante properties. Some previous similar examples include the random serial dictatorship, which is not stochastic-dominance efficient or envy-free (Bogomolnaia and Moulin, 2001), and DA with random tie-breaking, which is not ex-ante stable, and induces ex-ante discriminations (Kesten and Ünver, 2015).

domain. Decomposing the random allocations selected by PB also gives a deterministic mechanism that satisfies efficiency and respect for rank under weak preferences.

Similar to the original Boston mechanism, PB focuses on the allocation of agents' kth choices at each step k. Within a step, the allocation of probability shares of objects to agents follows a consumption procedure as in the prominent *probabilistic serial rule* (PS) that is introduced by Bogomolnaia and Moulin (2001) and extended to accommodate weak preferences by Katta and Sethuraman (2006). A defining feature of (extended) PS is *ordinal fairness* (Hashimoto et al., 2014, Heo and Yılmaz, 2015), which requires the allocation of each object to try to equalize any two agents' probabilities of receiving a weakly better object. We define *equal-rank ordinal fairness* by imposing the requirement only when the two agents rank the object equally. PB satisfies this new fairness axiom (which is stronger than equal treatment of equals), as well as sd-efficiency and respect for rank.

In the special case of strict preferences, the algorithm that defines PB is reduced to a simple procedure where at each step k each agent consumes her kth choice at the unit rate, starting from the time at which she previously stops consuming. The algorithm is more involved when preferences are weak. First, at a step k there could be more than enough objects to satisfy an agent's demand for her kth choices, and which shares of objects to allocate depend on other agents' ℓth choices, where $\ell > k$. We thus introduce a *guarantee* for the agent and defer the allocation to her to a later step. Second, within a step k, in light of the agents' nested kth indifference classes, we take a similar parametric network approach as in Katta and Sethuraman (2006), and first allocate objects to the *bottleneck* set of agents through a maximum flow.[4] Therefore, the procedure within each step generally consists of multiple rounds of allocation, with guarantees being possibly generated at the last round.

As the Boston mechanism and PS, PB is manipulable. We show that in general there does not exist a strategy-proof mechanism that satisfies our central requirement of respect for rank, even under strict preferences. However, replacing the Boston mechanism with random tie-breaking (which is only applicable to strict preferences) with PB is not as costly, since the former is also manipulable. This is in contrast to the trade-offs faced in considering probabilistic versions of other deterministic mechanisms, where a market designer has to choose between a strategy-proof one and a manipulable one with

[4]Some guarantees may also be delivered at the same time. Moreover, due to the network technique used, the algorithm finds the outcome in polynomial time.

better ex-ante properties.[5]

In the end, we briefly discuss a few closely related mechanisms. First, Han (2023) introduces a probabilistic version of the Boston mechanism for strict preferences and weak priorities. It is equivalent to PB when preferences are strict and priorities are degenerate.[6] Second, Chen et al. (2023) proposes and characterizes a similar mechanism that also satisfies sd-efficiency and respect for rank, in the context of strict preferences and degenerate priorities. We discuss the differences in details in Section 4.1. Finally, when there are outside options, the dichotomous preference domain becomes a special case of weak preferences. In this case, PB is equivalent to both the extended PS and the *egalitarian solution* of Bogomolnaia and Moulin (2004).

The rest of the paper is organized as follows. The next section presents the model. Section 3 introduces the Boston mechanism as well as PS, and motivates our design. In Section 4 we introduce PB, where we present its simple form for strict preferences and illustrate the mechanism for weak preferences through an example, before a formal definition. Section 5 discusses the properties of PB and Section 6 concludes. All proofs are given in the appendix.

2 The Model

Let N be a finite and non-empty set of **agents**, and A a set of **objects**. Assume $|N| = |A|$. Each agent $i \in N$ has a complete and transitive **preference relation** $\succsim_i$ on A, with $\succ_i$ and $\sim_i$ denoting its asymmetric and symmetric components respectively. A **preference profile** $\succsim = (\succsim_i)_{i \in N}$ is a list of individual preferences. We fix N and A in the rest of the paper. Then a **problem** is represented by a preference profile $\succsim$.[7]

[5]For instance, the probabilistic versions of DA (Kesten and Ünver, 2015, Han, 2023) and the top trading cycles mechanism (Yu and Zhang, 2021) are manipulable, while such mechanisms with random tie-breaking are strategy-proof.

[6]More specifically, Han (2023) focuses on a probabilistic version of DA that is generated through a method called "division". In general we can generate random mechanisms from deterministic mechanisms by this method. As an additional application, he shows that a probabilistic version of the Boston mechanism can be constructed under strict preferences. However, when preferences are weak, it is not clear how PB can be constructed from some deterministic mechanism by this method, given that the Boston mechanism is only defined for strict preferences.

[7]We focus on the restricted setup for simplicity. Since preferences are allowed to be weak, the mechanism we design can be easily extended to the more general model where $|N|$ and $|A|$ may not be equal and there are outside options by introducing dummy agents and dummy objects. Moreover, the case of many-to-one allocation, i.e., each object has multiple copies, can also be easily incorporated.

Given $\succsim$, for each agent $i \in N$, let $e_i(\succsim)$ be the number of indifference classes in her preferences, and $E_i^k(\succsim)$, where $k \in \{1, \ldots, e_i(\succsim)\}$, be the non-empty set of objects in her kth indifference class. We also set $E_i^k(\succsim) = \emptyset$ if $k > e_i(\succsim)$. Let $\bar{e}(\succsim) = \max\{e_i(\succsim) : i \in N\}$, and $E_M^k(\succsim) = \cup_{i \in M} E_i^k(\succsim)$ for any $M \subseteq N$ and $k \in \{1, \ldots, \bar{e}(\succsim)\}$. In addition, we use $r_i(\succsim, a)$ to denote the *rank* of object a in the preferences of i, i.e., $r_i(\succsim, a) = k$ if and only if $a \in E_i^k(\succsim)$.[8] When the problem under consideration is clear, we often omit the dependence of the above symbols on $\succsim$.

A **random allocation**, or simply an **allocation**, is represented by a $|N| \times |A|$ matrix P such that $P_{ia} \geq 0$, $\sum_{b \in A} P_{ib} = 1$, and $\sum_{j \in N} P_{ja} = 1$ for all $i \in N$ and $a \in A$, where P_{ia} is the probability that agent i is assigned object a. For each agent $i \in N$, let $P_i = (P_{ia})_{a \in A}$ denote the lottery obtained by i under the allocation P. An allocation P is **deterministic** if $P_{ia} \in \{0, 1\}$ for all $i \in N$ and $a \in A$. By the Birkhoff-von Neumann theorem (Birkhoff, 1946, von Neumann, 1953), every random allocation can be represented as a lottery over deterministic allocations.

Consider a problem $\succsim$. For any $i \in N$, $a \in A$ and allocation P, let $F(\succsim_i, a, P) = \sum_{b \in A : b \succsim_i a} P_{ib}$ be the probability that i is assigned an object weakly better than a under P. For two allocations P and P', each agent i can compare her lotteries P_i and P_i' using the first-order stochastic dominance relation $\succsim_i^{sd}$: $P_i \succsim_i^{sd} P_i'$ if $F(\succsim_i, a, P) \geq F(\succsim_i, a, P')$ for all $a \in A$. Then two allocations P and P' are **welfare equivalent** if $P_i \succsim_i^{sd} P_i'$ and $P_i' \succsim_i^{sd} P_i$ for all $i \in N$. An allocation P is **sd-efficient** if there does not exist another allocation P' such that (1) $P_i' \succsim_i^{sd} P_i$ for all $i \in N$, and (2) P and P' are not welfare equivalent. When a deterministic allocation is sd-efficient we simply say that it is efficient.

Given that all agents have equal claim to each object, using random allocations helps restore fairness. A minimum fairness requirement is that any two agents with the same preferences should receive lotteries that are equivalent in welfare terms. Formally, an allocation P satisfies **equal treatment of equals** if for any $i, j \in N$, $\succsim_i = \succsim_j$ implies $F(\succsim_i, a, P) = F(\succsim_j, a, P)$ for all $a \in A$.

Finally, a **mechanism** is a function φ that selects an allocation $\varphi(\succsim)$ to each problem $\succsim$. A mechanism is said to satisfy a certain axiom if its outcome satisfies the axiom for every problem. A mechanism φ is **strategy-proof** if for any $\succsim$, $i \in N$ and $\succsim_i'$, $\varphi_i(\succsim) \succsim_i^{sd} \varphi_i(\succsim_i', \succsim_{-i})$.

[8] For weak preferences there could be other definitions of "rank", and we focus on the simplest and natural one. This is also the rank concept adopted in the studies on rank-maximal matchings with weak preferences (see, for example, Irving et al. (2006)).

3 Motivation

The Boston mechanism has been mostly studied in the setting of priority-based allocation, or school choice problems, with strict preferences and strict priorities. To introduce the mechanism, we first envision that each object $a \in A$ has an exogenous priority ordering over the agents N. When the priority orderings are strict, for any strict preference profile $\succsim$ the Boston mechanism chooses a deterministic allocation through the following procedure:

> In each step $k \geq 1$, consider only the agents who have not been assigned an object, and the objects that have not been assigned to an agent. If the kth choice of agent i is object a, and i has higher priority for a than any other agent whose kth choice is a, then assign a to i.

The outcome is efficient, and respects preference ranks, or *favors higher ranks* (Kojima and Ünver, 2014), in the sense that if agent i envies the object a received by another agent j, then $r_j(\succsim, a) \leq r_i(\succsim, a)$.

If priorities are weak, as in many school choice programs in practice, a natural and fair solution is to pick an ordering of the agents from the uniform distribution to break the ties in priorities, and then apply the Boston mechanism. This leads to a random mechanism that satisfies the minimum fairness requirement of equal treatment of equals. However, as shown by Chen et al. (2023), the Boston mechanism with random tie-breaking is not sd-efficient, and does not respect preference ranks from the ex-ante perspective: the random allocation generated may fail to satisfy the following axiom.

Definition 1. *An allocation P for $\succsim$ satisfies **respect for rank** if there do not exist $i, j \in N$ and $a \in A$ such that $P_{ia} > 0$, $F(\succsim_j, a, P) < 1$, and $r_j(\succsim, a) < r_i(\succsim, a)$.*

By the definitions, it is straightforward to see that if a random allocation satisfies sd-efficiency and respect for rank, then for any lottery decomposition every deterministic allocation in the support is efficient and satisfies respect for rank.

Our goal is to design a fair random mechanism that satisfies sd-efficiency and respect for rank. We allow weak preferences, but restrict attention to the simple and special case of weak priorities where all agents have equal priority or claim for each object.[9]

[9] This can also model some school choice problems precisely: as discussed in Pathak and Sethuraman (2011), in the supplementary round of student assignment in New York City, all students have the same priority for each school.

Note that, for each problem $\succsim$, there is an endogenous and non-homogenous priority structure generated by agent preferences, where agent i has a higher priority than agent j at object a if $r_i(\succsim, a) < r_j(\succsim, a)$, and the axiom of respect for rank essentially requires a random allocation to be *ex-ante stable* (Roth et al., 1993, Kesten and Ünver, 2015) with respect to such priorities.

In designing a probabilistic version of the Boston mechanism, we will also incorporate features of the *probabilistic serial mechanism* (PS) from Bogomolnaia and Moulin (2001). Under strict preferences, it selects an allocation through a consumption (or "eating") procedure where the agents simultaneously consume their best available objects at the unit rate during the unit time interval. Hashimoto et al. (2014) show that *ordinal fairness* is a defining property of PS. An allocation P is ordinally fair for $\succsim$ if for any $i, j \in N$ and $a \in A$, $P_{ia} > 0$ implies $F(\succsim_j, a, P) \geq F(\succsim_i, a, P)$. It requires that the allocation of the probability shares of any object a should try to equalize everyone's surplus at a, i.e., the probability of receiving an object at least as good as a. Therefore, in general an agent with a smaller probability of receiving a strictly better object would receive a larger share of a, which is an obvious feature of PS as such agent starts to consume a earlier. As discussed in Han (2023), ordinal fairness can also be normatively justified through a general principle of fairness, *compensation*.[10] For the general case of weak preferences, Katta and Sethuraman (2006) generalize PS and introduce the *extended PS* solution, which is still characterized by ordinal fairness (Heo and Yılmaz, 2015).

Given that respect for rank guides the allocation of an object between agents who rank it differently, we can naturally impose the requirement of ordinal fairness for the agents who rank the object equally:

Definition 2. *An allocation P for $\succsim$ is* ***equal-rank ordinally fair*** *if for any $i, j \in N$ and $a \in A$ with $r_i(\succsim, a) = r_j(\succsim, a)$, $P_{ia} > 0$ implies $F(\succsim_j, a, P) \geq F(\succsim_i, a, P)$.*

It is straightforward to show that equal-rank ordinal fairness implies equal treatment of equals. Moreover, the Boston mechanism with random tie-breaking is not equal-rank ordinally fair.[11]

[10] Moulin (2004) provides detailed discussions of four principles of fairness: exogenous rights, compensation, reward, and fitness. In general, compensation says the allocation of resources should compensate for the differences in those primary individual characteristics, and equalize the shares of the higher-order characteristic or the more essential commodity. In considering the allocation of the shares of an object in our context, we take an agent's surplus at the object as the higher-order characteristic.

[11] One example is given by the problem in Example 2 below in Section 4.1. Let P' denote the outcome

Finally, it is worth mentioning that our three desiderata in the design, sd-efficiency, respect for rank and equal-rank ordinal fairness, are independent. Under strict preferences, the *fractional Boston rule* of Chen et al. (2023) only satisfies sd-efficiency and respect for rank, and PS only satisfies sd-efficiency and equal-rank ordinal fairness. More importantly, the following example illustrates that, under weak preferences, respect for rank in conjunction with equal-rank ordinal fairness does not imply sd-efficiency. This is in contrast to the fact that respect for rank itself implies sd-efficiency under strict preferences (Chen et al., 2023).[12]

Example 1. *Suppose that $N = \{1, 2, 3\}$ and $A = \{a, b, c\}$. Consider the following preferences $\succsim$, where agent 3's first choices are b and c, as well as an allocation P:*

$\succsim_1$	$\succsim_2$	$\succsim_3$
a	a	b, c
b	b	a
c	c	

$$P = \begin{array}{c|ccc} & a & b & c \\ \hline 1 & \frac{1}{2} & 0 & \frac{1}{2} \\ 2 & \frac{1}{2} & 0 & \frac{1}{2} \\ 3 & 0 & 1 & 0 \end{array}$$

It is easy to check that P satisfies respect for rank and equal-rank ordinal fairness. It is not sd-efficient as agent 2 can trade some shares of c for agent 3's shares of b.

4 The Probabilistic Boston Mechanism

4.1 The Case of Strict Preferences

We first present our mechanism in the simple case of strict preferences. As in the Boston mechanism, in each step k we only allocate agents' kth choices. Within each step, the allocation follows a consumption procedure as in PS. Overall, for each agent we will keep track of her own consumption timeline, and she will consume from $t = 0$ to $t = 1$ to receive a total probability shares of 1. Given a strict preference profile $\succsim$, the procedure can be simply described as follows:

> In each step $k \geq 1$, if agent i has consumed objects up to time $t \in [0, 1)$ before step k,[13] and her kth choice, an object a, is not exhausted yet, then

from applying the Boston mechanism after a single ordering is picked from the uniform distribution to break the ties at all objects. Then $P'_{3c} = \frac{31}{60}$ and $F(\succsim_3, c, P') = \frac{51}{60} > F(\succsim_1, c, P') = \frac{49}{60}$.

[12] For deterministic allocations this is first shown by Kojima and Ünver (2014).

[13] $t < 1$ indicates that she is not fully satisfied yet. Moreover, $t = 0$ if $k = 1$.

she starts to consume the remaining shares of a from t at the unit rate until it is exhausted or her consumption time reaches 1.

We use the following example to illustrate this procedure.

Example 2. *Suppose that $N = \{1,2,3,4,5\}$ and $A = \{a,b,c,d,e\}$. The preferences $\succsim$ and the allocation P selected by the mechanism are:*[14]

$\succsim_1$	$\succsim_2$	$\succsim_3$	$\succsim_4$	$\succsim_5$
a	a	b	b	b
c	d	c	a	a
d		d	d	d
e		e	e	e

$$P = \begin{array}{c|ccccc} & a & b & c & d & e \\ \hline 1 & \frac{1}{2} & 0 & \frac{5}{12} & 0 & \frac{1}{12} \\ 2 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 \\ 3 & 0 & \frac{1}{3} & \frac{7}{12} & 0 & \frac{1}{12} \\ 4 & 0 & \frac{1}{3} & 0 & \frac{1}{4} & \frac{5}{12} \\ 5 & 0 & \frac{1}{3} & 0 & \frac{1}{4} & \frac{5}{12} \end{array}$$

Step 1: 1 *and* 2 *consume a from $t = 0$ to $t = \frac{1}{2}$, and* 3, 4 *and* 5 *consume b from $t = 0$ to $t = \frac{1}{3}$.*

Step 2: 1 *and* 3 *consume their second choice c but they start from different times:* 1 *starts from $t = \frac{1}{2}$, while* 3 *starts from $t = \frac{1}{3}$. Then c is exhausted at $t = \frac{11}{12}$.* 2 *consumes d from $t = \frac{1}{2}$ to $t = 1$ and she is then fully satisfied.* 4 *and* 5 *do not consume as their second choice a is already exhausted in the previous step.*

Step 3: 1 *and* 3 *start to consume d from $t = \frac{11}{12}$, while* 4 *and* 5 *start to consume d from $t = \frac{1}{3}$. The remaining $\frac{1}{2}$ of d is exhausted at $t = \frac{7}{12}$ by* 4 *and* 5*, before* 1 *and* 3 *start.*

Step 4: 1*,* 3*,* 4 *and* 5 *consume e and they are fully satisfied after e is exhausted at $t = 1$.*

Under strict preferences our mechanism is similar to the fractional Boston rule of Chen et al. (2023). The difference is that in each step k of their rule the clock is reset for all agents and they start to consume their kth choices at the same time ($t = 0$). Therefore, this guarantees the key fairness desideratum of *equal-rank envy-freeness* in their design, which requires that the allocation of each object among the agents who rank it equally should try to equalize their probabilities of receiving this object. In contrast, in our case an agent always starts to consume from the time at which she stops in the last step, which ensures equal-rank ordinal fairness.

[14] For each preference relation we only list the first few objects that are relevant for the mechansim.

4.2 Preliminaries on Network Flows

In defining our mechanism for general weak preferences, we rely on network techniques. Below we first present the necessary concepts and results from network flow theory that will be used in the analysis.

A *directed graph* $(V, \mathscr{E})$ consists of a non-empty and finite set of vertices V and a set of edges $\mathscr{E} \subseteq \{(v, w) \in V \times V : v \neq w\}$. Then a *network* is a directed graph $(V, \mathscr{E})$ with a *source* vertex $s \in V$, a *sink* vertex $u \in V \setminus \{s\}$, and an edge capacity function $c : \mathscr{E} \to \mathbb{R}_+$. A *flow* in this network is a function $f : \mathscr{E} \to \mathbb{R}_+$ such that

1. $f(v, w) \leq c(v, w)$ for all $(v, w) \in \mathscr{E}$, and
2. $\sum_{w:(w,v)\in\mathscr{E}} f(w, v) = \sum_{w:(v,w)\in\mathscr{E}} f(v, w)$ for all $v \in V \setminus \{s, u\}$ (*conservation law*).

The *value* of the flow f, i.e., the amount sent from s to u, is

$$\sum_{v:(s,v)\in\mathscr{E}} f(s, v) - \sum_{v:(v,s)\in\mathscr{E}} f(v, s).$$ [15]

A *maximum flow* is a flow of maximum value.

On the other hand, a *cut* is represented by a set $X \subseteq V$ with $s \in X$ and $u \in V \setminus X$. The *capacity* of the cut X is

$$\sum_{(v,w)\in\mathscr{E}:v\in X, w\in V\setminus X} c(v, w).$$

With a slight abuse of notation, we use $c(X)$ to denote the capacity of the cut X. A *minimum cut* is a cut of minimum capacity. When both X and X' are minimum cuts, $X \cup X'$ is also a minimum cut.

By the conservation law, the value of a flow f is equal to the net flow across a cut X, i.e., $\sum_{(v,w)\in\mathscr{E}:v\in X, w\in V\setminus X} f(v, w) - \sum_{(v,w)\in\mathscr{E}:v\in V\setminus X, w\in X} f(v, w)$. Therefore, the value of a maximum flow cannot exceed the capacity of a minimum cut. Then, the central result in the network flow theory, the *max-flow min-cut theorem* (Ford and Fulkerson, 1956), states that the two numbers are equal. A key implication of the theorem that will be used later is that, given a minimum cut X and a maximum flow f, we have $f(v, w) = c(v, w)$ for all $(v, w) \in \mathscr{E}$ with $v \in X$ and $w \in V \setminus X$.

[15] By the conservation law this is equal to $\sum_{v:(v,u)\in\mathscr{E}} f(v, u) - \sum_{v:(u,v)\in\mathscr{E}} f(u, v)$.

4.3 Description of the Construction and an Illustrative Example

In the general weak preference domain we still focus on allocating agents' kth choices at the kth step, but there are two main complications in extending the simple allocation procedure in the case of strict preferences. First, at a step k there could be more than enough objects to fully satisfy an agent's demand for her kth choices. To achieve sd-efficiency, we have to defer the allocation of her kth choices to her to a later step. For instance, if an agent i has two first choices, a and b, and neither of them is a first choice of any other agent, then the shares of a and b allocated to i should depend on how these objects are desired at later steps. We handle this by introducing *guarantees*, i.e., let i be guaranteed a total shares of 1 from a and b. In general, guarantees generated at a step will always be delivered at a later step, which ensures respect for rank.

Second, when agents have multiple kth choices, and their kth choices (partially) overlap, different allocations at step k lead to different distributions of their probabilities of receiving the first k choices (i.e., their surpluses at their kth choices). To satisfy equal-rank ordinal fairness as well as sd-efficiency, we have to identify "bottleneck" set of agents, and, roughly speaking, allocate the most demanded objects first. Therefore, the allocation procedure within each step k will consist of multiple rounds. At each round, we take a similar approach as Katta and Sethuraman (2006), and construct a parametric network to find the bottleneck, allocate some agents' kth choices to them, and possibly deliver some guarantees (generated at previous steps) at the same time. In general, after several rounds of allocation, if at the last round there are more than enough objects to fully satisfy the remaining agents' demands for their kth choices, we generate guarantees for them.

Before formally defining the algorithm that finds the outcome of our mechanism, we construct a large problem to illustrate all the key aspects of the design. There are 11 agents and 11 objects: $N = \{1, \ldots, 11\}$ and $A = \{a_1, \ldots, a_{11}\}$. The preferences are:

$\succsim_1$	$\succsim_2$	$\succsim_3$	$\succsim_4$	$\succsim_5$	$\succsim_6$	$\succsim_7$	$\succsim_8$	$\succsim_9$	$\succsim_{10}$	$\succsim_{11}$
a_1, a_2	a_3	a_3	a_3	a_3	a_4	a_4	a_4	a_5	a_5	a_5
	a_1, a_2	a_1, a_6, a_7	a_4	a_4	a_3	a_3	a_8	a_8	a_3	a_3
			a_9	a_1, a_9	a_2, a_9	a_{10}	a_{10}	a_9, a_{10}	a_9, a_{10}	a_6, a_7, a_{10}
			a_7	a_7, a_{11}	a_{11}	a_{11}	a_{11}	a_{11}	a_{11}	

We will focus on explaining the several rounds within the third step of the algorithm, omitting the details and the networks for the first two steps.

At the first step, agents 2, 3, 4 and 5 are each assigned $\frac{1}{4}$ of a_3, and $6, \ldots, 11$ are each assigned $\frac{1}{3}$ of the first choice. Agent 1 is guaranteed a share of $g_1 = 1$ from the objects $D_1 = \{a_1, a_2\}$.

At the second step, 8 and 9 are each assigned $\frac{1}{2}$ of a_8. Given agent 1's guarantee, since 2 only demands $\frac{3}{4}$ from $\{a_1, a_2\}$ and 3 only demands $\frac{3}{4}$ from $\{a_1, a_6, a_7\}$, there are more than enough objects to fully satisfy the three agents, and the allocation of $\{a_1, a_2, a_6, a_7\}$ is not decisive at this point. Hence we create new guarantees: let $g_2 = \frac{3}{4}$, $D_2 = \{a_1, a_2\}$, $g_3 = \frac{3}{4}$ and $D_3 = \{a_1, a_6, a_7\}$. Besides, any other agent is not assigned anything as her second choice is already exhausted.

Below we summarize the information needed for the next step, where for an agent i without a guarantee, τ_i is the amount received so far and $D_i = E_i^3$.

i	1	2	3
g_i	1	$\frac{3}{4}$	$\frac{3}{4}$
D_i	a_1, a_2	a_1, a_2	a_1, a_6, a_7

i	4	5	6	7	8	9	10	11
τ_i	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{1}{3}$	$\frac{1}{3}$
D_i	a_9	a_1, a_9	a_2, a_9	a_{10}	a_{10}	a_9, a_{10}	a_9, a_{10}	a_6, a_7, a_{10}

Consider the third step. At the first round we construct the network in Figure 1, with a parameter λ. Each agent points to the objects that she demands, i.e., there is an edge from i to a if $a \in D_i$.[16] The capacity of such edge is set to ∞ (or some sufficiently large number) and we omit it for most of the edges. The source points to each agent i, and the capacity of (s, i) is g_i if i has a guarantee, and is $\max\{\lambda - \tau_i, 0\}$ otherwise. Each object a points to the sink, and the capacity of (a, u) is equal to 1. Therefore, when $\lambda \leq 1$, any flow f in the network gives a feasible assignment of (some of) the probability shares of the objects to the agents, where each i is assigned $f(i, a)$ of $a \in D_i$.

We gradually increase λ from the smallest τ_i, $\frac{1}{4}$. When λ is small and close to $\frac{1}{4}$, $\{s\}$ is always the unique minimum cut. In this case, the max-flow min-cut theorem indicates that, under any maximum flow, the flow on each (s, i) is equal to its capacity. That is, for a small λ we can allocate objects so that: (1) for each agent i without a guarantee, the

[16] In principle all the remaining objects are included in the network, but we omitted a_{11} as it is not demanded, and is thus irrelevant for the analysis.

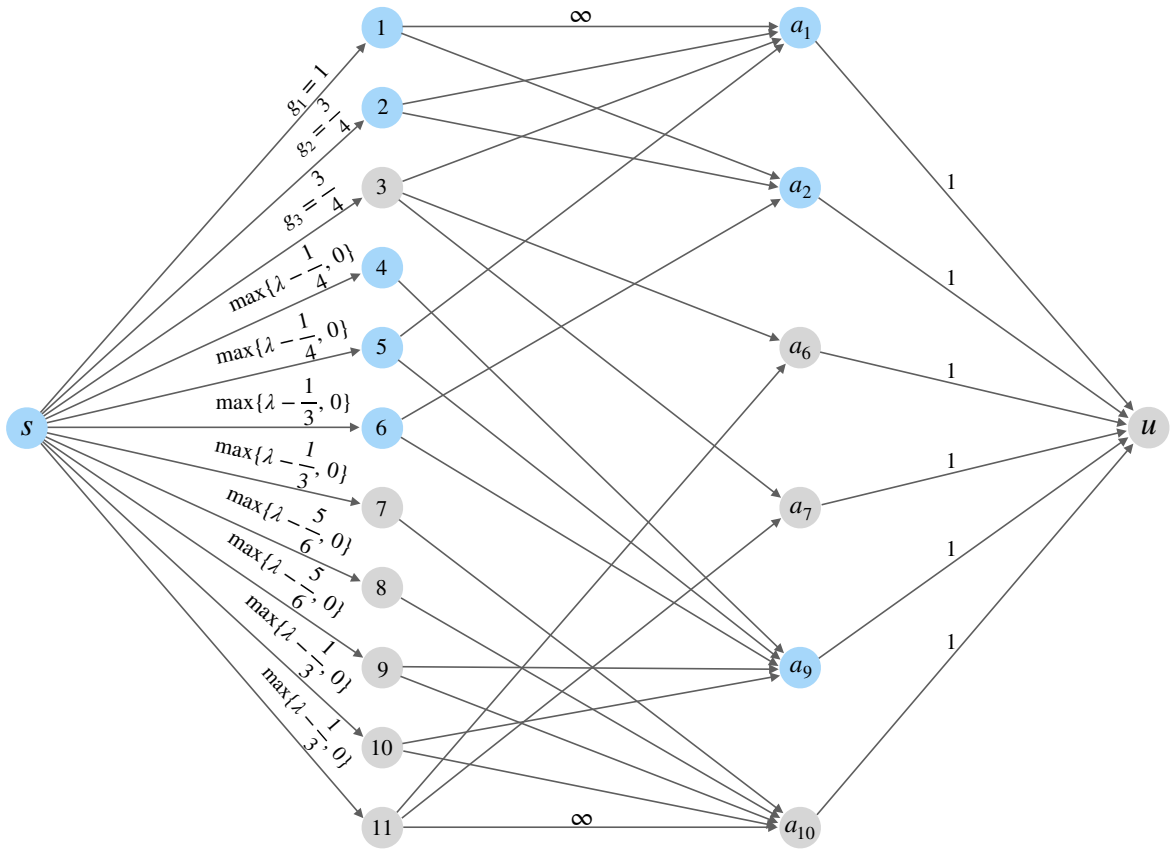

Figure 1: The network at round 1 of step 3. For $\lambda = \frac{25}{36}$ (the breakpoint) the largest minimum cut is $\{s, 1, 2, 4, 5, 6, a_1, a_2, a_9\}$.

probability of receiving her first three choices is $\max\{\lambda, \tau_i\}$, which ensures equal-rank ordinal fairness, and (2) all the guarantees are delivered at the same time.

When λ reaches the *breakpoint* of $\frac{25}{36}$, $\{s\}$ is still a minimum cut but no longer the unique one. At this point, $\{s, 1, 2, 4, 5, 6, a_1, a_2, a_9\}$ is the largest minimum cut, which includes the agents $1, 2, 4, 5, 6$ and all objects that they demand. In fact, under any maximum flow these objects are all assigned to them,[17] and the amount received by each i is equal to the capacity of (s, i). Therefore, we simply pick an arbitrary maximum flow to assign $\{a_1, a_2, a_9\}$ to the agents in the *bottleneck* set $\{1, 2, 4, 5, 6\}$: assign a share of 1 from $\{a_1, a_2\}$ to agent 1, $\frac{3}{4}$ from $\{a_1, a_2\}$ to 2, $\frac{25}{36} - \frac{1}{4}$ of a_9 to 4, $\frac{25}{36} - \frac{1}{4}$ from $\{a_1, a_9\}$ to 5, and $\frac{25}{36} - \frac{1}{3}$ from $\{a_2, a_9\}$ to 6.

The first round of step 3 then terminates, and the assigned agents become *inactive*

[17] We give a formal analysis of the network in the next section, and prove results such as this.

at the remaining rounds of this step. Note that a_9 is also demanded by agents 9 and 10 but they do not receive a share. The capacity of $(s, 9)$ is 0 at the breakpoint, as agent 9 already has a large probability of receiving her first two choices. On the other hand, as seen below agent 10 is assigned at the next round, where the breakpoint is larger, indicating a larger surplus at her third choices.

At the second round, we construct the network in Figure 2 with the active agents and remaining objects. At the breakpoint $\frac{5}{6}$, $\{s\}$ is a minimum cut, and $\{s, 7, 8, 9, 10, a_{10}\}$ is the largest minimum cut. The capacities of $(s, 8)$ and $(s, 9)$ are 0, and thus 8 and 9 do not receive any share in a flow. Hence we simply consider the *largest minimum cut without zero-capacity*, which is $\{s, 7, 10, a_{10}\}$. Then each one in the bottleneck set $\{7, 10\}$ is assigned $\frac{5}{6} - \tau_7 = \frac{1}{2}$ of a_{10}.

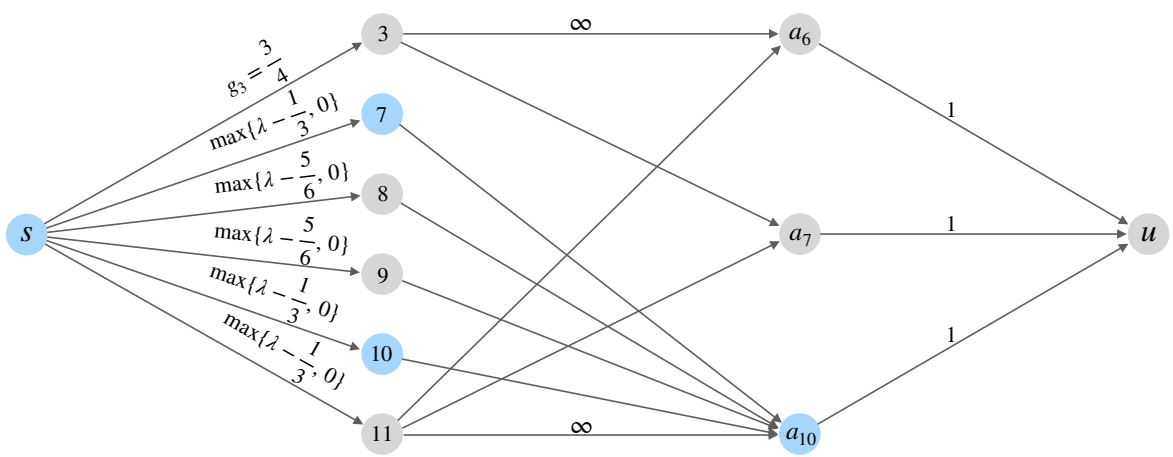

Figure 2: The network at round 2 of step 3. For $\lambda = \frac{5}{6}$ the largest minimum cut without zero-capacity is $\{s, 7, 10, a_{10}\}$.

At the third round, consider the network in Figure 3. Note that, although they are not assigned yet in step 3, agents 8 and 9 are no longer active as their third choices are exhausted. In this simple network, $\{s\}$ is still the unique minimum cut for $\lambda = 1$, i.e., there are more than enough objects to fully satisfy 3 and 11. Hence we generate a guarantee for 11 by setting $g_{11} = 1 - \tau_{11} = \frac{2}{3}$ and $D_{11} = \{a_6, a_7\}$. Then the procedure within step 3 terminates after round 3.

In the end, the fourth step is the final step with only one round of allocation, in which the breakpoint is equal to 1 and every agent is fully satisfied.

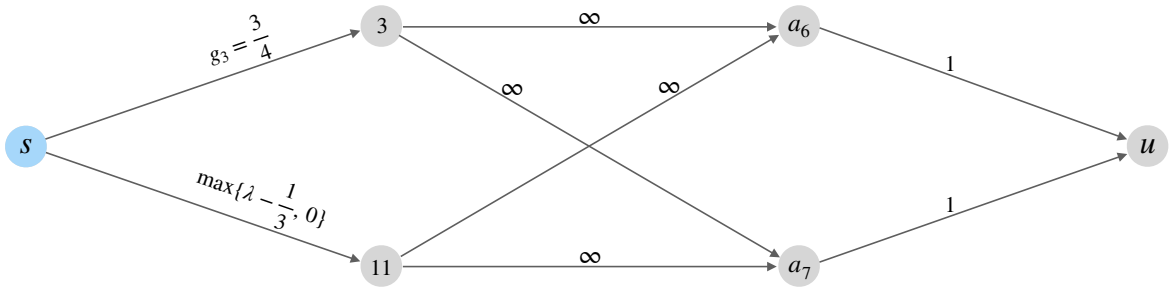

Figure 3: The network at round 3 of step 3. For $\lambda = 1$, $\{s\}$ is the unique minimum cut.

4.4 Analysis on the Parametric Network

In the remaining of Section 4, we fix a problem $\succsim$, and define the algorithm that finds the outcome of our mechanism. As the algorithm is rather involved, in this subsection we analyze and define how objects are allocated or guarantees are generated through the network approach at an arbitrary round. Then we present the complete algorithm that embeds the network approach in the following subsection.

Suppose that at round t of step k:

- The set of remaining objects is $A^* \neq \emptyset$, and the set of active agents is $N^* \neq \emptyset$.
- $I \subsetneq N^*$ is the set of active agents with guarantees. Each $i \in I$ is guaranteed $g_i \in (0, 1]$ from the objects $D_i \subseteq A^*$.
- For any non-empty $I' \subseteq I$, $\sum_{i \in I'} g_i < \left| \cup_{i \in I'} D_i \right|$. This ensures that there are more than enough objects to deliver all the guarantees.
- For each $i \in N^* \setminus I$, she has a non-empty set of available kth choices $D_i \subseteq A^*$, and the total probability shares assigned to her before step k is $\tau_i \in [0, 1)$.

For any $M \subseteq N^*$, let $D_M = \cup_{i \in M} D_i$ be the collection of objects demanded by the agents M. List the numbers $\left\{\tau_i : i \in N^* \setminus I\right\}$ as $\tau_1, \tau_2, \ldots, \tau_\ell$ such that $0 \leq \tau_1 < \tau_2 < \ldots < \tau_\ell < 1$. For any $\lambda \in [\tau_1, \bar{\lambda}]$, where $\bar{\lambda} = |A^*| + 1$, define a network $(V, \mathscr{E})$ with a source s, a sink u, and a capacity function c_λ such that

- $V = \{s\} \cup N^* \cup A^* \cup \{u\}$.
- $\mathscr{E} = \left\{(s, i) : i \in N^*\right\} \cup \left\{(i, a) : i \in N^*, a \in D_i\right\} \cup \left\{(a, u) : a \in A^*\right\}$.

- For each $i \in I$, $c_\lambda(s,i) = g_i$.
- For each $i \in N^* \setminus I$, $c_\lambda(s,i) = \max\{\lambda - \tau_i, 0\}$.
- For each $i \in N^*$ and $a \in D_i$, $c_\lambda(i,a) = \infty$.
- For each $a \in A^*$, $c_\lambda(a,u) = 1$.

To analyze the allocation of objects through maximum flows, the key is to understand the properties of minimum cuts in the network. Below we first present several technical results regarding minimum cuts.

4.4.1 Minimum Cuts

First, due to the infinite capacities of the edges from agents to objects, the objects included in a minimum cut are exactly the ones demanded by the agents in the cut:

Lemma 1. *For any $\lambda \in [\tau_1, \bar{\lambda}]$, if X is a minimum cut, then $X = \{s\} \cup M \cup D_M$ for some (possibly empty) $M \subseteq N^*$.*

Therefore, for brevity, given any $M \subseteq N^*$ we also use $\overline{M}$ to denote the cut $\{s\} \cup M \cup D_M$. The next lemma gives additional structural results on a minimum cut.

Lemma 2. *Suppose that $M \subseteq N^*$, $M \neq \emptyset$ and $\overline{M}$ is a minimum cut for some $\lambda \in [\tau_1, \bar{\lambda}]$. Then $M \setminus I \neq \emptyset$. Moreover, if $i \in M$ and $c_\lambda(s,i) = 0$, then $\overline{M \setminus \{i\}}$ is also a minimum cut.*

That is, it is not possible that all agents in a minimum cut $\overline{M}$ with $M \neq \emptyset$ have guarantees. Moreover, when the edge from s to an agent $i \in M$ has zero capacity, eliminating this agent leads to a smaller minimum cut. Therefore, for any λ, there is a minimum cut $\overline{M}$ such that $c_\lambda(s,i) > 0$ for all $i \in M$, which we refer to as a *minimum cut without zero-capacity*. Then by taking unions, there is a unique largest minimum cut without zero-capacity (in terms of set inclusion).

We are most interested in the circumstances under which $\{s\}$ is a minimum cut. Therefore, in the end we design a simple and intuitive procedure that finds the largest minimum cut without zero-capacity using an arbitrary maximum flow, when $\{s\}$ is a minimum cut. This will be useful in the following discussions on object allocation as well as computations.

Let $\lambda \in [\tau_1, \bar{\lambda}]$ and suppose that $\{s\}$ is a minimum cut. For any flow f, let $A(f) = \{a \in A^* : f(a,u) = 1\}$ denote the "fully-assigned" objects under f. The following *reduction procedure* constructs a sequence of maximum flows with decreasing sets of fully-assigned objects, starting from any maximum flow f_1.

At each step $m \geq 1$: if there are $i \in N^*$ and $a \in A(f_m)$ such that $f_m(i,a) > 0$ and $D_i \setminus A(f_m) \neq \emptyset$, then we construct another maximum flow f_{m+1} such that $A(f_{m+1}) = A(f_m) \setminus \{a\}$, by decreasing $f_m(i,a)$ and $f_m(a,u)$ by some small $\epsilon > 0$, and increasing $f_m(i,b)$ and $f_m(b,u)$ by ϵ for some $b \in D_i \setminus A(f_m)$;[18] the procedure stops at this step otherwise.

The procedure will stop at some step $\bar{m}$, and the outcome is the maximum flow $f_{\bar{m}}$.

Lemma 3. *Consider any $\lambda \in [\tau_1, \bar{\lambda}]$ such that $\{s\}$ is a minimum cut. Let f be the outcome of any reduction procedure, and*

$$M = \{i \in N^* : f(i,a) > 0 \text{ for some } a \in A(f)\}.$$

Then $\overline{M}$ is the largest minimum cut without zero-capacity.

It follows immediately from the above lemma that:

Corollary 1. *For any $\lambda \in [\tau_1, \bar{\lambda}]$, if $\{s\}$ is the unique minimum cut, then there is a maximum flow f with $A(f) = \emptyset$.*

4.4.2 Allocating the Objects

Let $\kappa(\lambda)$ be the capacity of a minimum cut for every $\lambda \in [\tau_1, \bar{\lambda}]$, i.e., κ is the *minimum cut capacity function*. Lemma 1 indicates that κ is the lower envelope of the finite set of functions $\{c_\lambda(\overline{M}) : M \subseteq N^*\}$, where for any M and λ,

$$c_\lambda(\overline{M}) = |D_M| + \sum_{i \in N^* \setminus M} c_\lambda(s,i).$$

It is straightforward to see that each $c_\lambda(\overline{M})$ is a piecewise linear (convex) function of λ.

[18] The procedure is iterative: even if we eliminate all such fully-assigned objects as a from $A(f_m)$ at this step, $A(f_{m+1})$ may still be reducible at the next step.

When $\lambda = \tau_1$, $c_\lambda(s,i) = 0$ for all $i \in N^* \setminus I$. It is clear that $\overline{M}$ cannot be a minimum cut if $M \neq \emptyset$ and $c_\lambda(s,i) = 0$ for all $i \in M$. Then it follows from Lemma 2 that $\{s\}$ is the unique minimum cut. Given the above discussions on κ, for a small $\epsilon > 0$, $\{s\}$ is also the unique minimum cut for $\lambda = \tau_1 + \epsilon$. Then, by the max-flow min-cut theorem, $f(s,i) = c_{\tau_1+\epsilon}(s,i)$ for any maximum flow f and $i \in N^*$.

Therefore, for a small λ we can allocate objects to the agents such that, for each $i \in N^* \setminus I$, her probability of receiving her first k choices is $\max\{\lambda, \tau_i\}$, and all the guarantees are delivered. Moreover, by Corollary 1, there are more than enough objects to achieve such allocation, i.e., no object would be exhausted. However, this may be infeasible for a larger λ.

When $\lambda = \bar{\lambda} = |A^*| + 1$, $c_\lambda\big(\{s\} \cup N^* \cup A^*\big) < c_\lambda\big(\{s\}\big)$, i.e., $\{s\}$ is no longer a minimum cut. Then, in light of the above discussions on κ being the lower envelope of $\big\{c_\lambda(\overline{M}) : M \subseteq N^*\big\}$, there exists $\lambda^* \in (\tau_1, \bar{\lambda})$ such that (1) for $\lambda \in [\tau_1, \lambda^*)$, $\{s\}$ is the unique minimum cut, and (2) for $\lambda = \lambda^*$, $\{s\}$ is a minimum cut but not the unique one, which indicates that it is not the largest minimum cut without zero-capacity. We refer to λ^* as the **breakpoint**. If $\lambda^* \leq 1$ and the largest minimum cut without zero-capacity for λ^* is $\overline{M}$, then $M \neq \emptyset$ is referred to as the **bottleneck**.

Consider the case that $\lambda = \lambda^* \leq 1$ and the bottleneck is M. Pick any maximum flow f. By the max-flow min-cut theorem, $f(s,i) = c_{\lambda^*}(s,i)$ for all $i \in N^*$. That is, we can still allocate the objects so that the probability of receiving her first k choices is $\max\{\lambda^*, \tau_i\}$ for each $i \in N^* \setminus I$, and all the guarantees are delivered. However, by the theorem again,

$$\sum_{i \in N^* \setminus M} f(s,i) + |D_M| = c_{\lambda^*}(\overline{M}) = c_{\lambda^*}(\{s\}) = \sum_{i \in N^* \setminus M} f(s,i) + \sum_{i \in M} f(s,i),$$

which implies $\sum_{i \in M} f(s,i) = |D_M|$, i.e., the objects D_M are exhausted and they are all allocated to the agents M under f. Furthermore, only the objects D_M need to be allocated at this point: if f' is the outcome of a reduction procedure, then it is straightforward to show $A(f') = D_M$ using Lemma 3.

Therefore, we let each $i \in M \setminus I$ be assigned $\lambda^* - \tau_i$ of the objects D_i, and each $i \in M \cap I$ be assigned g_i of the objects D_i, according to any maximum flow.

On the other hand, when $\lambda^* > 1$, $\{s\}$ is the unique minimum cut for $\lambda = 1$ and thus we can fully satisfy all the demands from N^* without exhausting any object. In this case we create new guarantees, instead of allocating objects: let each $i \in N^* \setminus I$ be

guaranteed $1-\tau_i$ from D_i.

4.4.3 Computations

As shown in the next subsection, the complete algorithm takes at most $\bar{e}(\succsim)$ steps for the problem $\succsim$, and within each step the number of rounds is bounded by the number of remaining objects plus one. For the parametric network at a given round, efficient methods to compute a maximum flow are well-understood, and our simple reduction procedure gives the bottleneck when the breakpoint $\lambda^* \leq 1$. Therefore, to have a polynomial-time algorithm, the only computational considerations are how to determine whether $\lambda^* \leq 1$, and to find the exact value of the breakpoint when $\lambda^* \leq 1$.

If $\tau_i = \tau_j$ for all $i, j \in N^* \setminus I$, then every edge capacity is a linear function of λ. In this special case, the minimum cut capacity function κ is a piecewise linear concave function, and a fast polynomial-time algorithm from Gallo et al. (1989) finds the smallest value of λ at which the slope of κ changes, which is the breakpoint λ^*.[19] While in general an edge capacity $c_\lambda(s,i)$ may not be linear in λ, we can still easily pin down λ^* using the the algorithm of Gallo et al. (1989) in conjunction with the reduction procedure.

Assume $\ell \geq 2$ and recall that $0 \leq \tau_1 < \tau_2 < \ldots < \tau_\ell < 1$. We first consider the interval $[\tau_1, \tau_2]$, and check if $\{s\}$ is a minimum cut for $\lambda = \tau_2$. If not, since each edge capacity is linear within the interval, the breakpoint $\lambda^* \in (\tau_1, \tau_2)$ can be found by the method of Gallo et al. (1989). On the other hand, when $\{s\}$ is a minimum cut for $\lambda = \tau_2$, we use the reduction procedure to find the largest minimum cut without zero-capacity, so that it can be determined if $\{s\}$ is the unique minimum cut. If not, $\lambda^* = \tau_2$. If $\{s\}$ is the unique minimum cut for $\lambda = \tau_2$, we repeat the above analysis for the next interval $[\tau_2, \tau_3]$ and continue in this fashion. In the end, if $\{s\}$ is the unique minimum cut for $\lambda = \tau_\ell$, we consider the interval $[\tau_\ell, 1]$, and will either find $\lambda^* \in (\tau_\ell, 1]$ or conclude that $\lambda^* > 1$.

4.5 The Complete Algorithm

At each step $k \geq 1$:

- The set of remaining objects is A^k, and the set of remaining agents is N^k.

[19] Recall that κ is the lower envelope of the functions $\{c_\lambda(\overline{M}) : M \subseteq N^*\}$, each of which is linear on $[\tau_1, \bar{\lambda}]$ in this case.

- $I^k \subseteq N^k$ is the set of remaining agents with guarantees. Each $i \in I^k$ is guaranteed $g_i \in (0,1]$ from the objects $G_i \cap A^k$.
- For any non-empty $I' \subseteq I^k$, $\sum_{i \in I'} g_i < \left| \cup_{i \in I'} (G_i \cap A^k) \right|$.
- The total probability shares assigned to each $i \in N^k \setminus I^k$ before step k is $\tau_i^k \in [0,1)$.

Remark 1. *Suppose $A^k \neq \emptyset$. As will be seen from the construction of the algorithm, the total remaining demands must be equal to the number of remaining objects, i.e., $\sum_{i \in N^k \setminus I^k}(1 - \tau_i^k) + \sum_{i \in I^k} g_i = |A^k|$. In conjunction with the above inequality regarding guarantees, this indicates that $N^k \setminus I^k \neq \emptyset$.*

Let $A^1 = A$, $N^1 = N$, $I^1 = \emptyset$, and $\tau_i^1 = 0$ for all $i \in N^1$.

Within step k, at each round $t \geq 1$:

- The set of remaining objects is A_t^k.
- The set of active agents is N_t^k. We consider an agent $i \in N^k$ active at round t if (1) $i \notin I^k$ and some of her kth choices are still available at this round, or (2) $i \in I^k$ and her guarantee has not been delivered.
- $I_t^k \subseteq N_t^k$ is the set of active agents with guarantees. For any non-empty $I' \subseteq I_t^k$, $\sum_{i \in I'} g_i < \left| \cup_{i \in I'} (G_i \cap A_t^k) \right|$.

Let $A_1^k = A^k$, $N_1^k = \left\{ i \in N^k \setminus I^k : E_i^k \cap A^k \neq \emptyset \right\} \cup I^k$, and $I_1^k = I^k$. For each $i \in N^k \setminus N_1^k$, her kth choices are already exhausted, and we set $\tau_i^{k+1} = \tau_i^k$ before round 1.

At round t, if $N_t^k \setminus I_t^k \neq \emptyset$, setup the parametric network as defined before, and denote the breakpoint as λ_t^k. If $\lambda_t^k \leq 1$, find the value of λ_t^k as well as the bottleneck M_t^k. Allocate the objects $\left\{ E^k_{M_t^k \setminus I_t^k} \cup \left\{ \cup_{i \in M_t^k \cap I_t^k} G_i \right\} \right\} \cap A_t^k$ such that:

- each $i \in M_t^k \setminus I_t^k$ is assigned $\lambda_t^k - \tau_i^k$ of the objects in $E_i^k \cap A_t^k$, and
- each $i \in M_t^k \cap I_t^k$ is assigned g_i of the objects in $G_i \cap A_t^k$.

Then, after allocation, we continue to the next round by defining

- $\tau_i^{k+1} = \lambda_t^k$ for each $i \in M_t^k \setminus I_t^k$,
- $A_{t+1}^k = A_t^k \setminus \left\{ E^k_{M_t^k \setminus I_t^k} \cup \left\{ \cup_{i \in M_t^k \cap I_t^k} G_i \right\} \right\}$,

- $I^k_{t+1} = I^k_t \setminus M^k_t$,
- $N^k_{t+1} = \left\{ i \in N^k_t \setminus \left\{ I^k_t \cup M^k_t \right\} : E^k_i \cap A^k_{t+1} \neq \emptyset \right\} \cup I^k_{t+1}$, and
- $\tau^{k+1}_i = \tau^k_i$ for each $i \in N^k_t \setminus \left\{ N^k_{t+1} \cup M^k_t \right\}$.[20]

The procedure within step k terminates after round t if (1) $N^k_t \setminus I^k_t = \emptyset$, or (2) $N^k_t \setminus I^k_t \neq \emptyset$ and $\lambda^k_t > 1$. Then, we continue to the next step by defining

- $A^{k+1} = A^k_t$,
- $I^{k+1} = N^k_t$,
- for each $i \in I^{k+1} \setminus I^k$, $G_i = E^k_i \cap A^k_t$ and $g_i = 1 - \tau^k_i$, and
- $N^{k+1} = \left\{ i \in N^k \setminus \left\{ I^k \cup I^{k+1} \right\} : \tau^{k+1}_i < 1 \right\} \cup I^{k+1}$.

Finally, the algorithm terminates after step k if $A^{k+1} = \emptyset$.

The following lemma ensures that the assumption on guarantees at the beginning of each round or step is always satisfied, and hence the algorithm is well-defined.

Lemma 4. *At any round t of any step k, $\sum_{i \in I'} g_i < \left| \cup_{i \in I'} (G_i \cap A^k_t) \right|$ for any non-empty $I' \subseteq I^k_t$.*

On the termination of the algorithm, the following result shows that all objects are allocated after at most $\bar{e}$ steps, and thus the outcome of the algorithm is a well-defined random allocation.

Lemma 5. *There is $k \in \{1, \ldots, \bar{e}\}$ such that $A^{k+1} = \emptyset$.*

In general, there can be multiple ways to allocate the objects at a given round, but any two outcome allocations from the algorithm are welfare equivalent. Therefore, we simply define the **probabilistic Boston mechanism** (PB) as some function that selects one outcome allocation from the algorithm for every possible problem.

[20] At round t, such an agent is active, does not have a guarantee, and all of her available kth choices are allocated to other agents.

5 Properties

We formally show that the three initial desiderata discussed in Section 3 are satisfied by PB.

Theorem 1. *The probabilistic Boston mechanism satisfies sd-efficiency, respect for rank, and equal-rank ordinal fairness.*

The property of respect for rank is easy to see, since we allocate kth choices to agents as much as possible at each step k, and all guarantees are delivered in the end. Overall, the objects are allocated sequentially at multiple rounds of multiple steps. As established in the proof, sd-efficiency follows from the fact that, for any round of allocation, each assigned agent receives shares from her currently best available objects that are also all exhausted at this round. Finally, equal-rank ordinal fairness is guaranteed by the construction of each network, as well as the key property that breakpoints are increasing within each step.

Regarding incentives, PB is not a strategy-proof mechanism. For instance, as the original Boston mechanism, under PB an agent may have the incentive to push up the rank of some less preferred object in face of competition with other agents at her top-ranked objects. Formalizing this simple intuition, we show that the central requirement of respect for rank in our study is in general incompatible with strategy-proofness, even in the strict preference domain.

Theorem 2. *If $|N| \geq 3$ and every possible preference relation is strict, then there does not exist a strategy-proof mechanism that satisfies respect for rank.*[21]

In addition, it is already known from Bogomolnaia and Moulin (2001) that the other main requirement, sd-efficiency, cannot be achieved by any strategy-proof mechanism that satisfies equal treatment of equals when $|N| \geq 4$ and preferences are strict.

As mentioned at the beginning, PB can be easily extended to the more general allocation model where $|N|$ and $|A|$ may not be equal and there are outside options. In the special case of dichotomous preferences, i.e., each agent finds any object either acceptable (better than her outside option) or unacceptable (worse than her outside option), and is indifferent between acceptable objects, the first step of the algorithm is sufficient to determine the outcome of PB, which is equivalent to the extended PS and the egalitarian solution of Bogomolnaia and Moulin (2004).

[21] It is easy to see that PB is strategy-proof if $|N| = 2$.

On the other hand, outside options can make a difference in impossibility results. Consider the more general allocation model with strict preferences. An allocation is *individually rational* if no agent receives an unacceptable object with positive probability. It is *non-wasteful* if no agent has a positive probability of receiving an outcome worse than an object that is not fully assigned. Then the simple proof of Theorem 2 can be easily modified to show that, if $|N| \geq 3$ and $|A| \geq 2$, there does not exist a non-wasteful and strategy-proof mechanism that satisfies respect for rank. Moreover, Martini (2016) provides a strong impossibility: if $|N| \geq 4$ and $|A| \geq 3$, then there does not exist an individually rational, non-wasteful and strategy-proof mechanism that satisfies equal treatment of equals.

6 Conclusion

In this study we design a probabilistic version of the Boston mechanism to restore its attractive properties from the ex-ante perspective, by incorporating the essence of the probabilistic serial rule. While the algorithm that defines PB involves complicated details, the outcome is easy to compute through the network approach, and the properties of PB are intuitive and straightforward to convey. Given that the Boston mechanism with random tie-breaking is also manipulable, a transition to PB does not seem to cost much. At the same time, the transition improves the welfare and fairness of the system, and also makes it possible for agents to express indifferences in preferences.

Appendix

A Proofs

A.1 Proofs of Results in Section 4

Proof of Lemma 1. Given any $\lambda \in [\tau_1, \bar{\lambda}]$, suppose that X is a minimum cut. Then $X = \{s\} \cup M \cup A'$ for some $M \subseteq N^*$ and $A' \subseteq A^*$. Since $c_\lambda(i, a) = \infty$ for all $i \in N^*$ and $a \in D_i$, we have $D_M \subseteq A'$. By the construction of the network,

$$c_\lambda(X) = |A'| + \sum_{i \in N^* \setminus M} c_\lambda(s, i).$$

If $A' \neq D_M$, then $D_M \subsetneq A'$ and

$$c_\lambda\big(X \setminus \{A' \setminus D_M\}\big) = |D_M| + \sum_{i \in N^* \setminus M} c_\lambda(s,i) < c_\lambda(X),$$

contradicting to X being a minimum cut. Therefore, we have $A' = D_M$, and $X = \{s\} \cup M \cup D_M$. □

Proof of Lemma 2. Let $\overline{M}$ be a minimum cut for $\lambda \in [\tau_1, \bar{\lambda}]$ with $M \neq \emptyset$.

First, suppose that $M \subseteq I$. By the initial assumption on guarantees, $\sum_{i \in M} g_i < |D_M|$. Then

$$c_\lambda\big(\overline{M}\big) = |D_M| + \sum_{i \in N^* \setminus M} c_\lambda(s,i) > \sum_{i \in M} g_i + \sum_{i \in N^* \setminus M} c_\lambda(s,i) = c_\lambda\big(\{s\}\big).$$

Hence $\{s\}$ is a cut with a smaller capacity, and a contradiction is reached.

Second, suppose that $i \in M$ and $c_\lambda(s,i) = 0$. Then

$$c_\lambda\big(\overline{M \setminus \{i\}}\big) = |D_{M \setminus \{i\}}| + \sum_{j \in N^* \setminus \{M \setminus \{i\}\}} c_\lambda(s,j) \leq |D_M| + c_\lambda(s,i) + \sum_{j \in N^* \setminus M} c_\lambda(s,j) = c_\lambda\big(\overline{M}\big).$$

This shows that $\overline{M \setminus \{i\}}$ is also a minimum cut. □

Proof of Lemma 3. Let λ, f and M be specified as in the statement of the lemma. We first show that $\overline{M}$ is a minimum cut. By the construction of the reduction procedure, it is straightforward to see that $A(f) = D_M$. Since there do not exist $i \in N^* \setminus M$ and $a \in D_M$ such that $f(i,a) > 0$, by the conservation law,

$$\sum_{i \in M} f(s,i) = \sum_{a \in D_M} f(a,u),$$

which is also equal to $|D_M|$, as $f(a,u) = 1$ for all $a \in D_M$. Then we have

$$\begin{aligned} c_\lambda\big(\overline{M}\big) &= |D_M| + \sum_{i \in N^* \setminus M} c_\lambda(s,i) \\ &= \sum_{i \in M} f(s,i) + \sum_{i \in N^* \setminus M} c_\lambda(s,i) \\ &\leq \sum_{i \in M} c_\lambda(s,i) + \sum_{i \in N^* \setminus M} c_\lambda(s,i) \\ &= c_\lambda\big(\{s\}\big). \end{aligned}$$

Given that $\{s\}$ is a minimum cut, $\overline{M}$ is a minimum cut.

For each $i \in M$, since $f(i,a) > 0$ for some $a \in D_i$, $c_\lambda(s,i) > 0$. That is, $\overline{M}$ is a minimum cut without zero-capacity. Assume to the contrary, $M \subsetneq M'$ and $\overline{M'}$ is also a minimum cut without zero-capacity. Then $D_M \subseteq D_{M'}$. By the max-flow min-cut theorem, $f(a,u) = 1$ for all $a \in D_{M'}$. This implies $D_{M'} \subseteq A(f) = D_M$ and thus $D_{M'} = D_M$. Then, $M \neq \emptyset$, and

$$c_\lambda\big(\overline{M'}\big) = |D_M| + \sum_{i \in N^* \setminus M'} c_\lambda(s,i) < |D_M| + \sum_{i \in N^* \setminus M} c_\lambda(s,i) = c_\lambda\big(\overline{M}\big),$$

where the inequality follows from the fact that $c_\lambda(s,i) > 0$ for all $i \in M' \setminus M$. This contradicts to $\overline{M}$ being a minimum cut. □

Proof of Lemma 4. The inequality condition on guarantees is clearly satisfied at the first round of the first step as $I_1^1 = \emptyset$. To prove by induction, suppose that it holds at some round t of some step k. There are two possible cases to consider.

Case 1: round t is the last round of step k, and the algorithm does not terminate after step k.

If $N_t^k \setminus I_t^k = \emptyset$, then $I_t^k = N_t^k = I^{k+1} = I_1^{k+1}$ and clearly the condition on guarantees holds at the first round of step $k+1$.

Suppose that $N_t^k \setminus I_t^k \neq \emptyset$ and new guarantees are generated at this round. By construction, $I^{k+1} = N_t^k$, $A^{k+1} = A_t^k$, $G_i = E_i^k \cap A_t^k$ and $g_i = 1 - \tau_i^k$ for all $i \in I^{k+1} \setminus I_t^k$. Consider the parametric network constructed at round t of step k, where the breakpoint $\lambda_t^k > 1$. For $\lambda = 1$, $\{s\}$ is the unique minimum cut. Therefore, for any non-empty $I' \subseteq I^{k+1}$ we have

$$c_1\big(\overline{I'}\big) = \big|\cup_{i \in I'} (G_i \cap A^{k+1})\big| + \sum_{i \in I^{k+1} \setminus I'} c_1(s,i) > c_1\big(\{s\}\big) = \sum_{i \in I'} g_i + \sum_{i \in I^{k+1} \setminus I'} c_1(s,i).$$

It follows that $\sum_{i \in I'} g_i < \big|\cup_{i \in I'} (G_i \cap A^{k+1})\big|$. Then such condition on guarantees also holds at the first round of step $k+1$.

Case 2: round t is not the last round of step k.

Then some objects are allocated at this round. Assume to the contrary, there is non-empty $I' \subseteq I_{t+1}^k$ such that $\sum_{i \in I'} g_i \geq \big|\cup_{i \in I'} (G_i \cap A_{t+1}^k)\big|$. Consider the network constructed

at round t. We have $I' \subseteq I_t^k$, $M_t^k \cap I' = \emptyset$, and

$$\begin{aligned}
c_{\lambda_t^k}\big(\overline{M_t^k \cup I'}\big) &= \left|\left\{E^k_{M_t^k \setminus I_t^k} \cup \left\{\cup_{i \in \{M_t^k \cap I_t^k\} \cup I'} G_i\right\}\right\} \cap A_t^k\right| + \sum_{i \in N_t^k \setminus \{M_t^k \cup I'\}} c_{\lambda_t^k}(s,i) \\
&= \left|\left\{E^k_{M_t^k \setminus I_t^k} \cup \left\{\cup_{i \in M_t^k \cap I_t^k} G_i\right\}\right\} \cap A_t^k\right| + \left|\cup_{i \in I'} (G_i \cap A_{t+1}^k)\right| + \sum_{i \in N_t^k \setminus \{M_t^k \cup I'\}} c_{\lambda_t^k}(s,i) \\
&\leq \left|\left\{E^k_{M_t^k \setminus I_t^k} \cup \left\{\cup_{i \in M_t^k \cap I_t^k} G_i\right\}\right\} \cap A_t^k\right| + \sum_{i \in I'} g_i + \sum_{i \in N_t^k \setminus \{M_t^k \cup I'\}} c_{\lambda_t^k}(s,i) \\
&= c_{\lambda_t^k}\big(\overline{M_t^k}\big).
\end{aligned}$$

This contradicts to the fact that $\overline{M_t^k}$ is the largest minimum cut without zero-capacity for λ_t^k. □

Proof of Lemma 5. Assume to the contrary, $A^{\bar{e}+1} \neq \emptyset$. Then in light of Remark 1, we can find $i \in N^{\bar{e}+1} \setminus I^{\bar{e}+1}$ and $a \in A^{\bar{e}+1}$ such that $a \in E_i^k$ for some $k \leq e_i \leq \bar{e}$. As $\tau_i^{\bar{e}+1} < 1$, $\tau_i^k < 1$ and $i \in N^k$. Suppose that the procedure within step k terminates after round t. As a is not allocated to any agent within step k, i is always active within step k and $i \in N_t^k$. By construction, this implies $i \in I^{k+1}$. Then agent i remains to be an agent with a guarantee until her guarantee is delivered. Therefore, it is impossible that $i \in N^{\bar{e}+1} \setminus I^{\bar{e}+1}$. □

A.2 Proof of Theorem 1

Pick any problem $\succsim$ and let P be the allocation selected by the probabilistic Boston mechanism. Consider the procedure in our algorithm. The following simple fact will be used in the proof of each property.

Lemma 6. *If $a \in A^k$ and $r_i(a) < k$, then $F(\succsim_i, a, P) = 1$.*

Proof. Let $a \in A^k$ and $r_i(a) = k' < k$. If $i \notin N^{k'} \setminus I^{k'}$, then clearly $F(\succsim_i, a, P) = 1$. Suppose that $i \in N^{k'} \setminus I^{k'}$, and round t is the last round of step k'. As a is available at each round of step k', agent i is active at each round of step k'. In particular, $i \in N_t^{k'} = I^{k'+1}$. That is, a guarantee is generated for i at round t of step k'. It follows that $F(\succsim_i, a, P) = 1$. □

This lemma immediately implies that P satisfies respect for rank: for any $i, j \in N$ and $a \in A$ such that $P_{ia} > 0$ and $r_j(a) < r_i(a)$, agent i must be assigned a share of a

at step $r_i(a)$ or a later step, and thus $a \in A^{r_i(a)}$, which indicates $F(\succsim_j, a, P) = 1$ by the lemma.

Next, to establish sd-efficiency, we show that whenever some objects are allocated to some agents at some round, each of these agents is assigned shares of her best available objects, and all of her best available objects are allocated at the same time.

Lemma 7. *Suppose that an agent $i \in M_t^k$ is assigned a share of $a \in A_t^k$ at round t of step k. Then $a \succsim_i b$ for all $b \in A_t^k$. Moreover, if $c \in A_t^k$ and $a \sim_i c$, then c is also allocated at this round.*

Proof. Suppose that i is assigned a share of a at round t of step k. Then $r_i(a) \leq k$ (with strict inequality when $i \in I_t^k$). If there is $b \in A_t^k$ such that $b \succ_i a$, $r_i(b) < k$. Then by Lemma 6, $F(\succsim_i, b, P) = 1$, which contradicts to $P_{ia} > 0$.

Let $c \in A_t^k$ and $a \sim_i c$. If $i \notin I_t^k$, then $r_i(a) = k$ and the objects $E_i^k \cap A_t^k$ are all allocated at this round. Hence c is also allocated. On the other hand, suppose that $i \in I_t^k$ and the guarantee of i is generated at the last round, round t', of step $k' < k$. Then $G_i = E_i^{k'} \cap A_{t'}^{k'}$, which includes both a and c. It follows that c is also allocated at round t of step k, as the objects $G_i \cap A_t^k$ are all allocated at this round. □

Assume to the contrary, P is not sd-efficient. By a characterization of sd-efficiency in Katta and Sethuraman (2006) (Lemma 2 in their paper), we can find Pareto-improving exchanges of probability shares among some agents at P: formally, there exists a list of objects $(a_1, a_2, \ldots, a_n)$ such that $n \geq 2$ and the following conditions are satisfied.

- For each $k \in \{1, \ldots, n\}$, there exists $i_k \in N$ such that $a_k \succsim_{i_k} a_{k+1}$ and $P_{i_k a_{k+1}} > 0$, where $a_{n+1} = a_1$.
- There is $\ell \in \{1, \ldots, n\}$ such that $a_\ell \succ_{i_\ell} a_{\ell+1}$.

Then Lemma 7 implies that a_ℓ is allocated before $a_{\ell+1}$ (i.e., a_ℓ is allocated at an earlier round of the same step or at an earlier step), and for each k, a_k and a_{k+1} are allocated at the same round or a_k is allocated before a_{k+1}. Hence a contradiction is reached.

Finally, we show equal-rank ordinal fairness, which relies crucially on the fact that breakpoints are increasing within a step:

Lemma 8. *If there are objects allocated at both round t and round $t+1$ of step k, then $\lambda_t^k < \lambda_{t+1}^k$.*

Proof. Assume to the contrary, there are objects allocated at both round t and round $t+1$ of step k, and $\lambda_t^k \geq \lambda_{t+1}^k$. Consider the network at round $t+1$ first. Since both $\{s\}$ and $\overline{M_{t+1}^k}$ are minimum cuts for $\lambda = \lambda_{t+1}^k$, their capacities are equal, which implies

$$\sum_{i \in M_{t+1}^k} c_{\lambda_{t+1}^k}(s,i) = \left| \left\{ E^k_{M_{t+1}^k \setminus I_{t+1}^k} \cup \left\{ \cup_{i \in M_{t+1}^k \cap I_{t+1}^k} G_i \right\} \right\} \cap A_{t+1}^k \right|. \tag{1}$$

Then, consider the network at round t. We have

$$\begin{aligned}
c_{\lambda_t^k}\left(\overline{M_t^k \cup M_{t+1}^k}\right) &= \left| \left\{ E^k_{M_t^k \setminus I_t^k} \cup \left\{ \cup_{i \in M_t^k \cap I_t^k} G_i \right\} \right\} \cap A_t^k \right| \\
&\quad + \left| \left\{ E^k_{M_{t+1}^k \setminus I_{t+1}^k} \cup \left\{ \cup_{i \in M_{t+1}^k \cap I_{t+1}^k} G_i \right\} \right\} \cap A_{t+1}^k \right| + \sum_{i \in N_t^k \setminus \{M_t^k \cup M_{t+1}^k\}} c_{\lambda_t^k}(s,i) \\
&= \left| \left\{ E^k_{M_t^k \setminus I_t^k} \cup \left\{ \cup_{i \in M_t^k \cap I_t^k} G_i \right\} \right\} \cap A_t^k \right| \\
&\quad + \left| \left\{ E^k_{M_{t+1}^k \setminus I_{t+1}^k} \cup \left\{ \cup_{i \in M_{t+1}^k \cap I_{t+1}^k} G_i \right\} \right\} \cap A_{t+1}^k \right| - \sum_{i \in M_{t+1}^k} c_{\lambda_t^k}(s,i) + \sum_{i \in N_t^k \setminus M_t^k} c_{\lambda_t^k}(s,i) \\
&\leq \left| \left\{ E^k_{M_t^k \setminus I_t^k} \cup \left\{ \cup_{i \in M_t^k \cap I_t^k} G_i \right\} \right\} \cap A_t^k \right| + \sum_{i \in N_t^k \setminus M_t^k} c_{\lambda_t^k}(s,i) \\
&= c_{\lambda_t^k}\left(\overline{M_t^k}\right),
\end{aligned}$$

where the inequality follows from equation (1) and the fact that $c_{\lambda_t^k}(s,i) \geq c_{\lambda_{t+1}^k}(s,i)$ for all $i \in M_{t+1}^k$. In addition, as $\overline{M_{t+1}^k}$ is a minimum cut without zero-capacity for λ_{t+1}^k in the network at round $t+1$, $c_{\lambda_t^k}(s,i) \geq c_{\lambda_{t+1}^k}(s,i) > 0$ for all $i \in M_{t+1}^k$. Therefore, $c_{\lambda_t^k}\left(\overline{M_t^k \cup M_{t+1}^k}\right) \leq c_{\lambda_t^k}\left(\overline{M_t^k}\right)$ implies that $\overline{M_t^k}$ is not the largest minimum cut without zero-capacity for λ_t^k in the network at round t, and a contradiction is reached. □

Consider any $i, j \in N$ and $a \in A$ with $r_i(a) = r_j(a) = k$ and $P_{ia} > 0$. If i is assigned a share of a at step $k' > k$, then by Lemma 6 $F(\succsim_j, a, P) = 1 \geq F(\succsim_i, a, P)$. We also have $F(\succsim_j, a, P) = 1$ if $j \notin N^k \setminus I^k$. Therefore, in the remaining proof we assume that i is assigned a share of a at round t of step k, and $i, j \in N^k \setminus I^k$.

Consider the objects $E_j^k \cap A^k$, where $a \in E_j^k \cap A^k \neq \emptyset$. If $E_j^k \cap A^{k+1} \neq \emptyset$, i.e., the remaining kth choices of j are not all allocated within step k, then by Lemma 6 $F(\succsim_j, a, P) = 1$. Suppose that the kth choices of j are exhausted after the allocation at round t' of step k. That is, $E_j^k \cap A_{t'}^k \neq \emptyset$ and $E_j^k \cap A_{t'+1}^k = \emptyset$. Then $t' \geq t$ as i is assigned a share of a at round t, and $j \in N_{t'}^k$ by construction. We consider two final scenarios.

1. $j \notin M^k_{t'}$. In the network at round t', since $E^k_j \cap A^k_{t'}$ is a subset of the set of objects allocated to $M^k_{t'}$ and $\overline{M^k_{t'}}$ is a minimum cut for $\lambda^k_{t'}$, it is straightforward to see that $c_{\lambda^k_{t'}}(s, j) = 0$. Therefore,

$$F(\succsim_j, a, P) \geq \tau^k_i \geq \lambda^k_{t'} \geq \lambda^k_t = F(\succsim_i, a, P),$$

where the last inequality follows from Lemma 8.

2. $j \in M^k_{t'}$. Then by Lemma 8,

$$F(\succsim_j, a, P) = \lambda^k_{t'} \geq \lambda^k_t = F(\succsim_i, a, P).$$

This finishes the proof of equal-rank ordinal fairness.

A.3 Proof of Theorem 2

Suppose that $|N| \geq 3$, preferences are strict, and a mechanism φ satisfies respect for rank. Pick three distinct agents i, j, k, and two objects a and b. Consider a preference profile $\succsim$ such that the first choices of the three agents are a, their second choices are b, and any other agent's first choice is not b. Then, for some $\ell \in \{i, j, k\}$, $\varphi_{\ell a}(\succsim) + \varphi_{\ell b}(\succsim) < 1$. If agent ℓ reports b as her first choice in $\succsim'_\ell$, then by respect for rank $\varphi_{\ell b}(\succsim'_\ell, \succsim_{-\ell}) = 1$, i.e., φ is not strategy-proof.

References

ABDULKADIROĞLU, A. AND T. SÖNMEZ (2003): "School choice: A mechanism design approach," *American Economic Review*, 93, 729–747.

AGARWAL, N. AND P. SOMAINI (2018): "Demand analysis using strategic reports: An application to a school choice mechanism," *Econometrica*, 86, 391–444.

BIRKHOFF, G. (1946): "Three observations on linear algebra," *Univ. Nac. Tucumán. Revista A*, 5, 147–151.

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

——— (2004): "Random matching under dichotomous preferences," *Econometrica*, 72, 257–279.

CALSAMIGLIA, C., C. FU, AND M. GÜELL (2020): "Structural estimation of a model of school choices: The Boston mechanism versus its alternatives," *Journal of Political Economy*, 128, 642–680.

CHEN, Y., P. HARLESS, AND Z. JIAO (2023): "The fractional Boston random assignment rule and its axiomatic characterization," *Review of Economic Design*, 1–23.

CHEN, Y. AND O. KESTEN (2017): "Chinese college admissions and school choice reforms: A theoretical analysis," *Journal of Political Economy*, 125, 99–139.

DOĞAN, B. AND B. KLAUS (2018): "Object allocation via immediate-acceptance: characterizations and an affirmative action application," *Journal of Mathematical Economics*, 79, 140–156.

FORD, L. R. AND D. R. FULKERSON (1956): "Maximal flow through a network," *Canadian Journal of Mathematics*, 8, 399–404.

GALE, D. AND L. S. SHAPLEY (1962): "College admissions and the stability of marriage," *American Mathematical Monthly*, 69, 9–15.

GALLO, G., M. D. GRIGORIADIS, AND R. E. TARJAN (1989): "A fast parametric maximum flow algorithm and applications," *SIAM Journal on Computing*, 18, 30–55.

HAN, X. (2023): "A theory of fair random allocation under priorities," *Working Paper*.

HASHIMOTO, T., D. HIRATA, O. KESTEN, M. KURINO, AND M. U. ÜNVER (2014): "Two axiomatic approaches to the probabilistic serial mechanism," *Theoretical Economics*, 9, 253–277.

HE, Y. (2017): "Gaming the Boston school choice mechanism in Beijing," *Working Paper*.

HEO, E. J. AND Ö. YILMAZ (2015): "A characterization of the extended serial correspondence," *Journal of Mathematical Economics*.

HYLLAND, A. AND R. ZECKHAUSER (1979): "The efficient allocation of individuals to positions," *Journal of Political Economy*, 91, 293–314.

IRVING, R. W., T. KAVITHA, K. MEHLHORN, D. MICHAIL, AND K. E. PALUCH (2006): "Rank-maximal matchings," *ACM Transactions on Algorithms (TALG)*, 2, 602–610.

KATTA, A.-K. AND J. SETHURAMAN (2006): "A solution to the random assignment problem on the full preference domain," *Journal of Economic theory*, 131, 231–250.

KESTEN, O. AND M. U. ÜNVER (2015): "A theory of school choice lotteries," *Theoretical Economics*, 10, 543–595.

KOJIMA, F. AND M. U. ÜNVER (2014): "The "Boston" school choice mechanism: an axiomatic approach," *Economic Theory*, 55, 515–544.

MARTINI, G. (2016): "Strategy-proof and fair assignment is wasteful," *Games and Economic Behavior*, 98, 172–179.

MOULIN, H. (2004): *Fair division and collective welfare*, MIT press.

PATHAK, P. A. AND J. SETHURAMAN (2011): "Lotteries in student assignment: An equivalence result," *Theoretical Economics*, 6, 1–17.

ROTH, A. E., U. G. ROTHBLUM, AND J. H. VANDE VATE (1993): "Stable matchings, optimal assignments, and linear programming," *Mathematics of operations research*, 18, 803–828.

TERRIER, C., P. A. PATHAK, AND K. REN (2021): "From immediate acceptance to deferred acceptance: effects on school admissions and achievement in England," Tech. rep., National Bureau of Economic Research.

VON NEUMANN, J. (1953): "A certain zero-sum two-person game equivalent to the optimal assignment problem," *Contributions to the Theory of Games*, 2, 5–12.

YU, J. AND J. ZHANG (2021): "Efficient and fair trading algorithms in market design environments," *arXiv preprint arXiv:2005.06878*.