UTMD-057

Strategyproof Mechanism for Two-Sided Matching with Resource Allocation

Kwei-guu Liu, Kentaro Yahiro, Makoto Yokoo
Kyushu University

January 5, 2023

Artificial Intelligence 316 (2023) 103855

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint

Strategyproof Mechanism for Two-Sided Matching with Resource Allocation ☆

Kwei-guu Liu, Kentaro Yahiro *, Makoto Yokoo

Kyushu University, Fukuoka, Japan

ARTICLE INFO

Article history:
Received 9 February 2022
Received in revised form 30 December 2022
Accepted 1 January 2023
Available online 5 January 2023

Keywords:
Matching
Resource allocation
Strategyproofness

ABSTRACT

In this work, we consider a student-project-resource matching-allocation problem, where students have preferences over projects and the projects have preferences over students. In this problem, students and indivisible resources are many-to-one matched to projects whose capacities are endogenously determined by the resources allocated to them. Traditionally, this problem is decomposed into two separate problems: (1) resources are allocated to projects based on expectations (a resource allocation problem), and (2) students are matched to projects based on the capacities determined in the previous problem (a matching problem). Although both problems are well-understood, if the expectations used in the first are incorrect, we obtain a sub-optimal outcome. Thus, this problem should be solved as a whole without dividing it into two parts. We show that no strategyproof mechanism satisfies fairness and weak efficiency requirements. Given this impossibility result, we develop a new class of strategyproof mechanisms called Sample and Deferred Acceptance (SDA), which satisfies several properties on fairness and efficiency. We experimentally compare several SDA instances as well as existing mechanisms, and show that an SDA instance strikes a good balance of fairness and efficiency when students are divided into different types according to their preferences.

1. Introduction

We introduce a simple, but fundamental problem called Student-Project-Resource matching-allocation problem (SPR).[1] On one hand, SPR can be considered as a two-sided, many-to-one matching problem [3] since students are matched to projects based on their preferences. On the other hand, it is also a discrete resource allocation problem [4] since resources are allocated to each project. However, unlike the standard setting of two-sided many-to-one matching, where the capacity of each project is exogenously determined, we assume the capacities are endogenously determined by resource allocation.

☆ This paper is based on our conference paper [1], which introduces just one mechanism called Sample and Vote Deferred Acceptance (SVDA). In this paper, we present a new class of mechanisms which we call Sample and Deferred Acceptance (SDA), and show axiomatic properties for this general class (SVDA corresponds to SDA-V in this paper). Furthermore, we develop other SDA instances (SDA-V* and SDA-S), and conduct an extended evaluation of SDA instances, including a case when students are divided into several types (Section 5).

* Corresponding author.
E-mail addresses: liu@agent.inf.kyushu-u.ac.jp (K.-g. Liu), yahiro@agent.inf.kyushu-u.ac.jp (K. Yahiro), yokoo@inf.kyushu-u.ac.jp (M. Yokoo).

[1] An SPR is a strict generalization of the student-project-room allocation problem [2], in which a single resource (i.e., a room) is allocated to each project while multiple resources can be allocated to each project in our model.

https://doi.org/10.1016/j.artint.2023.103855

If the mechanism designer knows the preferences of the students, she can allocate resources to projects using combinatorial optimization techniques. If each project's capacity is determined, even if the mechanism designer does not know the students' preferences beforehand, she can find a matching that satisfies desirable properties (e.g., stability) with a strategyproof mechanism, e.g., Deferred Acceptance mechanism (DA) [5], such that students voluntarily disclose their true preferences. However, the mechanism designer usually does not know their preferences. Thus, a common practice for solving this problem is to determine its resource allocation part based on the expectations or the past data and set the capacities of the projects. Then the actual matching of students to projects is determined by a matching mechanism. In this approach, if the expectations used in the first problem are incorrect, the outcome can be sub-optimal; excess demand and supply for seats can coexist, which can be avoided by a better resource allocation.

One real-life instance where this practice is applied is the nursery-school waiting list problem [6]. As of October 2018, over 47,000 children were on waiting lists for nursery schools in Japan. This serious social problem shackles women's empowerment. The Japanese government is trying to increase the number of nursery schools to encourage women to enter the workforce. The following is the procedure for matching children and teachers to publicly certified nursery schools in Japanese municipalities. First, the matching authority announces the quotas for each age group. This situation can be formalized as an SPR by assuming a child is a *student*, each age group in a school is a *project*, and a teacher is a *resource*.[2] Allocation of the resources/teachers within a school to each age group is based on estimates. Next, based on the quotas for each age group, the actual assignment is determined by a matching mechanism. The primary shortcoming of this procedure is that in the obtained matching, excess supply and demand may coexist in one school because the local authorities must determine the quotas for each age group of all the schools before they know the actual demand. To avoid such inefficiency, this problem should be solved as a whole without dividing it into two parts.

Another example is a school choice program for assigning students to public schools. In a standard setting, each school has a maximum quota that is determined in advance. Assume a local government (e.g., a city/prefecture/state) has spare resources, e.g., sufficient budget to hire temporary teachers, which can be allocated based on the demand. Then the maximum quota of each school is no longer fixed in advance, but it can be flexibly modified based on the actual demand utilizing extra budget/resources.

This paper follows previous works that address constrained matching problems. Two-sided matching has attracted considerable attention from AI and theoretical computer science researchers [7–13]. A standard market deals with maximum quotas, i.e., capacity limits that cannot be exceeded. However, many real-world matching markets are subject to a variety of distributional constraints [14,15], including regional maximum quotas, which restrict the total number of students assigned to a set of schools [16], minimum quotas, which guarantee that a certain number of students are assigned to each school [17–21], and diversity constraints [22–26]. Other works examine the computational complexity of finding a matching with desirable properties under distributional constraints [27–29]. One desirable property a matching should satisfy is stability [30–32], which can be divided into fairness and nonwastefulness, where nonwastefulness is related to students' welfare. When some distributional constraints are imposed, a stable matching may not exist. Given this fact, one stream of works considers weaker stability requirements [33–35,16,36], while another stream assumes fairness as hard constraints and tries to maximize students' welfare, e.g., by utilizing mixed integer linear programming (MILP) [37,34]. Although a model that resembles ours is examined [38], it utilizes a compact representation scheme to handle exponentially many students, assuming they can be divided into a small number of types.

Several works have addressed three-sided matching problems [39–41] where three types of players/agents are matched, e.g., males, females, and pets. Three-dimensional (or multi-dimensional) roommate problems deal with forming triplets (or d-tuples where $d \geq 3$) of agents with a single type [42,43]. Although these models superficially resemble ours, they are fundamentally different. In our model, a resource is not an agent/player; it has no preference over projects/students. A project/student has no preference over resources; a project just needs to receive sufficient resources to accommodate the students who have applied to it. In the student-project allocation problem [44], students are matched to projects, each of which is offered by a lecturer. A student has a preference over projects, and a lecturer has a preference over students. Each lecturer has a capacity limit. This problem can be considered as a standard two-sided matching problem with distributional constraints [14]. Aziz et al. [34] consider a problem called summer internship problem, where students are allocated to projects. The capacity of a project is endogenously determined by the amount of the budget that a supervisor provides. Although this problem resembles an SPR, one important difference is that we assume a resource is indivisible, while they assume the budget is infinitely divisible. Furthermore, they do not consider strategyproof mechanisms.

In our problem, the indivisibility of resources among projects leads to the computational difficulty when finding a feasible matching. Such hardness is also studied in the problem matching with sizes and its closely related problem parallel machine scheduling [45,46]. One typical problem of matching with sizes is known as matching with couples, where entities on one side have sizes of one or two people and the other side has resource capacities of two. If couples can be separated, the problem can be solved efficiently because it becomes the classical college admission problem. Due to the constraint that couples are inseparable in real-world problems like resident doctors matching schemes, the difficulty has been proved to be NP-complete [45].

[2] This nursery school problem has a distinct structure, i.e., a child/student must be assigned to one age group/project within a nursery school, and a resource/teacher is shared within the projects/age groups that belong to the same nursery school. Our formalization of an SPR is more general and can represent a variety of applications beyond this example.

In this work, we present a generalized framework to capture two orthogonal problems that need to be solved simultaneously as a single problem. We show that no strategyproof mechanism satisfies fairness (i.e., no student has justified envy) and very mild efficiency requirements on students' welfare [1]. We also confirm the limitations of the following three existing mechanisms [2]: Serial Dictatorship mechanism (SD), Artificial Caps Deferred Acceptance mechanism (ACDA), and Adaptive Deferred Acceptance mechanism (ADA). Then we introduce a new class of strategyproof mechanisms called Sample and Deferred Acceptance (SDA), which satisfies several properties on fairness and efficiency. SDA can be considered as a combination of SD and DA. Although it borrows a common idea from auction mechanisms, i.e., dividing students/participants into two groups and utilizing the information obtained by one group to appropriately set parameters to apply the mechanism to another group, its application to two-sided matching is novel. Moreover, we believe that combining SD and DA, such that the entire mechanism satisfies several desiderata, is unprecedented.

More specifically, in SDA, students are divided into two groups: *sampled students* and *regular students*. Sampled students are assigned using SD. Next, resources are allocated based on the preferences of the sampled students. Then the regular students are assigned using DA based on the resource allocation. We can use several alternative methods for determining the resource allocation based on the sampled students' preferences. One simple method is using the vote among sampled students. We call an instance of SDA using a simple Borda count *SDA with Voting (SDA-V)*. However, such a simple voting scheme can be inappropriate when students are divided into several types based on their preferences. To address this issue, we develop another instance of SDA, which uses a more sophisticated voting scheme called *SDA-V**. Furthermore, we develop another instance of SDA, which uses a simulation instead of voting called *SDA with Simulation (SDA-S)*. We experimentally show that SDA-S outperforms other SDA variants in terms of efficiency and fairness when students are divided into several types.

This paper is organized as follows. Section 2 introduces a model of an SPR. Section 3 shows that no mechanism is fair, weakly nonwasteful, resource efficient, and strategyproof. In Section 4, we explain the existing strategyproof mechanisms and their drawbacks, and introduce our new class of mechanisms, SDA. In Section 5, we show that SDA strikes a good balance between fairness and efficiency when all students belong to a single type. Next, we show that SDA-S outperforms other variants of SDA when students are divided into multiple types. Finally, Section 6 concludes our work.

2. Model

We define a Student-Project-Resource matching-allocation problem (SPR) as follows:

Definition 1 *(Student-Project-Resource allocation (SPR) instance).* An SPR instance is a tuple $(S, P, R, \succ_S, \succ_P, T_R, q_R)$.

- $S = \{s_1, \ldots, s_{|S|}\}$ is a set of students.
- $P = \{p_1, \ldots, p_{|P|}\}$ is a set of projects.
- $R = \{r_1, \ldots, r_{|R|}\}$ is a set of indivisible resources.
- $\succ_S = (\succ_s)_{s\in S}$ are the student strict preferences over set $P \cup \{\emptyset\}$. Symbol $\emptyset$ means that a student is not assigned to any project.
- $\succ_P = (\succ_p)_{p\in P}$ are the projects' strict preferences over set $S \cup \{\emptyset\}$. Symbol $\emptyset$ means that a project is assigned no student.
- $q_R = (q_r)_{r\in R}$ are the capacities of resources; $q_r \in \mathbb{N}_{>0}$ for every $r \in R$.
- $T_R = (T_r)_{r\in R}$ is a profile of the resource compatibility lists, where each $T_r \subseteq P$ is a set of projects to which resource r can be allocated. Since resource r is indivisible, it must be allocated to exactly one project in T_r.

Note that since we assume a resource indivisible, we cannot allocate a resource r with the capacity of two to the projects p_1 and p_2 with the capacity of one each. We illustrate our setting with the following example.

Example 1. There are four students, s_1, s_2, s_3, s_4, four projects, p_1, p_2, p_3, p_4, and two resources, r_1, r_2, where $T_{r_1} = \{p_1, p_2\}$, $T_{r_2} = \{p_3, p_4\}$, and $q_{r_1} = 2$, $q_{r_2} = 1$. The following are their preferences:

$$
\begin{aligned}
&s_1 : p_1 \succ p_2 \succ p_4 \succ p_3 \succ \emptyset, \quad && p_1 : s_4 \succ s_3 \succ s_2 \succ s_1 \succ \emptyset,\\
&s_2 : p_1 \succ p_2 \succ p_3 \succ p_4 \succ \emptyset, \quad && p_2 : s_4 \succ s_3 \succ s_2 \succ s_1 \succ \emptyset,\\
&s_3 : p_1 \succ p_2 \succ p_3 \succ p_4 \succ \emptyset, \quad && p_3 : s_1 \succ s_2 \succ s_3 \succ s_4 \succ \emptyset,\\
&s_4 : p_4 \succ p_3 \succ p_2 \succ \emptyset \succ p_1, \quad && p_4 : s_1 \succ s_2 \succ s_3 \succ s_4 \succ \emptyset.
\end{aligned}
$$

Since we assume resources are indivisible, it is impossible to allocate students to three different projects although the total capacity of all the resources equals three. A resource can only be allocated to a compatible project, e.g., r_1 can be allocated to either p_1 or p_2. The following are the possible capacities of the four projects: $(2, 0, 1, 0)$, $(2, 0, 0, 1)$, $(0, 2, 1, 0)$, or $(0, 2, 0, 1)$.

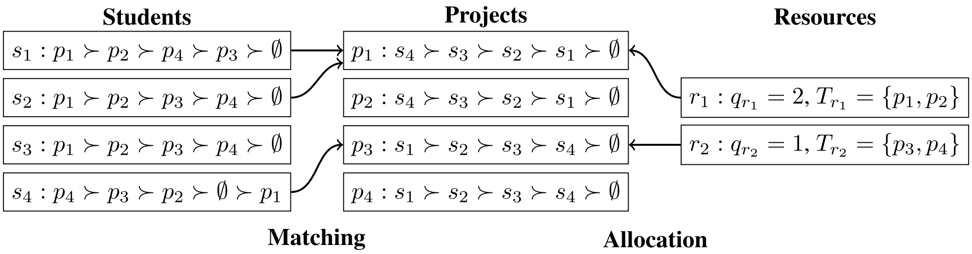


Fig. 1. SPR instance: matching $\hat{Y}$ and allocation $\hat{\mu}$ in Example 1.

We follow the matching with a contracts model [47]. Contract $(s, p) \in S \times P$ means that student s is matched to project p. Contract (s, p) is acceptable to student s (resp. project p) if $p \succ_s \emptyset$ holds (resp. $s \succ_p \emptyset$). Let X denote the set of all contracts that are acceptable to the projects.[3]

A matching is a set of contracts satisfying the following conditions.

Definition 2 *(Matching).* A matching is a subset $Y \subseteq X$, where for every student $s \in S$, $Y_s = \{(s, p) \mid (s, p) \in X\}$, either $|Y_s| = 0$, or $Y_s = \{(s, p)\}$ and $p \succ_s \emptyset$ hold.

For matching Y, let $Y(s)$ denote the project to which s is matched ($Y(s) = \emptyset$ if s is not matched to any project in Y), and let $Y(p) \subseteq S$ denote the set of students assigned to project p ($Y(p) = \emptyset$ means no student is allocated to p in Y).

In an SPR, we also need to describe how resources are allocated to projects. A matching's *feasibility* is defined based on this description.

Definition 3 *(Allocation).* An allocation $\mu : R \to P$ maps each resource r to a project $\mu(r) \in T_r$. Let $q_\mu(p) = \sum_{r \in \mu^{-1}(p)} q_r$.[4]

In words, a project's maximum quota is defined by the sum of all resources that are allocated to it. Note that multiple resources can be allocated to a single project.

Definition 4 *(Feasibility).* A feasible matching (Y, μ) is a matching-allocation pair where $|Y(p)| \le q_\mu(p)$ holds for every $p \in P$.

In the setting of Example 1, assume matching $\hat{Y}$ is $\{(s_1, p_1), (s_2, p_1), (s_3, \emptyset), (s_4, p_3)\}$ and allocation $\hat{\mu}$ distributes r_1 to p_1, and r_2 to p_3. Then $(\hat{Y}, \hat{\mu})$ is a feasible matching. See Fig. 1 for an illustration.

Next, we introduce a concept related to efficiency called *nonwastefulness*. First, we define a situation where a student claims that the current matching is inefficient since her welfare can be improved without disadvantaging other students.

Definition 5 *(Claiming an empty seat with μ).* Given feasible matching (Y, μ), student s claims an empty seat in project p with μ if the following conditions hold:

- $p \succ_s Y(s)$, and
- $Y \setminus \{(s, Y(s))\} \cup \{(s, p)\}$ is feasible with μ.

In other words, student s claims an empty seat in project p with μ if it is possible to move her to p from current project $Y(s)$ (which can be $\emptyset$) with current allocation μ.

Definition 6 *(Nonwastefulness).* Given feasible matching (Y, μ), student s *possibly* claims an empty seat in project p if $\exists \mu'$ such that s claims an empty seat in p with μ'. A feasible matching (Y, μ) is nonwasteful if no student possibly claims an empty seat in any project.

In other words, s possibly claims an empty seat in p if s can be moved to a more preferred project p without changing the assignment of the other students with allocation μ'. Note that μ' can be different from μ. Thus, s can possibly claim an empty seat in p even if it is impossible to move her to p with current allocation μ, as long as it becomes possible with a different/better allocation μ'. In a traditional setting, since the maximum quota (capacity limit) of each project is fixed, it

---

[3] In designing a strategyproof mechanism, we assume student preferences are private information, and the other information is public. Thus, we assume X, i.e., the set of all contracts that are acceptable to the projects is public.

[4] For $\mu^{-1}(p) = \emptyset$, we assume $q_\mu(p) = 0$.

suffices to check whether a student can be moved to another project under the fixed maximum quotas. In contrast, in our setting, maximum quotas are endogenous and flexible. Thus, the definition of nonwastefulness is modified to reflect this flexibility.

In the setting of Example 1 in Fig. 1, s_4 cannot claim an empty seat in p_4 in current allocation $\hat{\mu}$ because no resource is allocated to p_4 and it is impossible to move her from p_3 to p_4. However, she possibly claims an empty seat in p_4 since by allocating r_2 to p_4, we can move her to p_4 without disadvantaging other students. Thus, $(\hat{Y}, \hat{\mu})$ does not satisfy nonwastefulness.

Next, we introduce a concept called *fairness*.

Definition 7 *(Fairness).* Given feasible matching (Y, μ), student s has justified envy toward student s' if for project p such that $s' \in Y(p)$, $p \succ_s Y(s)$ and $s \succ_p s'$ hold. A feasible matching (Y, μ) is fair if no student has justified envy.

In other words, student s has justified envy toward s' if s' is assigned to project p although s prefers p over her current project $Y(s)$ and project p also prefers s over s'.

In the setting of Example 1 in Fig. 1, s_3 has justified envy toward s_1 (or s_2) since she prefers p_1 over $\emptyset$, and p_1 prefers her over s_1 (or s_2).

By combining nonwastefulness and fairness, we obtain *stability*.

Definition 8 *(Stability).* A feasible matching (Y, μ) is stable if it is nonwasteful and fair.

This stability concept is also called *strong stability* [16,48]. Next, we introduce concepts related to efficiency and welfare of students.

Definition 9 *(Pareto efficiency).* Matching Y is Pareto dominated by Y' if all students weakly prefer Y' over Y (that is, either $Y'(s) \succ_s Y(s)$ or $Y(s) = Y(s')$ for every $s \in S$) and at least one student strictly prefers Y'. Matching Y is strongly Pareto dominated by Y' if all students strictly prefer Y' over Y. A feasible matching is Pareto efficient if no feasible matching Pareto dominates it. A feasible matching is weakly Pareto efficient if no feasible matching strongly Pareto dominates it.

If a matching is Pareto efficient, we need to sacrifice the welfare of other students to improve the assignment of one student. If a matching is weakly Pareto efficient, it is impossible to strictly improve the assignments of all the students. Pareto efficiency obviously implies weak Pareto efficiency but not vice versa. Pareto efficiency also implies nonwastefulness since if a matching is wasteful, i.e., student s possibly claims an empty seat in project p, then we can move s to p from her current assignment without changing the welfare of other students using appropriate allocation μ'. The converse is not true. Weak Pareto efficiency and nonwastefulness are independent properties.

In the setting of Example 1, matching $\hat{Y}$ in Fig. 1 is not Pareto efficient since s_4 possibly claims an empty seat in p_4; we can improve the assignment of s_4 without disadvantaging other students. On the other hand, it is weakly Pareto efficient since s_1 and s_2 are assigned to their best project and their assignment cannot be improved.

Next, we formally define a mechanism and introduce the desirable properties a mechanism should satisfy: strategyproofness and weak group strategyproofness.

Definition 10 *(Mechanism).* Given any SPR instance, a mechanism outputs a feasible matching (Y, μ). If a mechanism always yields a feasible matching that satisfies property A (e.g., fairness), we say that this mechanism is A (e.g., fair).

Definition 11 *((Group) strategyproofness).* A mechanism is strategyproof if no student has an incentive to misreport her preference. A mechanism is weakly group strategyproof if no group of students can collude to misreport their preferences in a way that makes every member strictly better off.

3. Impossibility theorems

In an SPR, resources should be flexibly allocated to projects for more efficient matching. However, such flexibility is hard to combine with fairness. First, we show that fairness and nonwastefulness are incompatible.

Theorem 1. *An SPR instance exists where no feasible matching is fair and nonwasteful.*

Proof. Consider the following SPR instance[5]: two students, s_1, s_2, two projects, p_1, p_2, and a unitary resource compatible with both. The student preferences are $p_1 \succ_{s_1} p_2 \succ_{s_1} \emptyset$, and $p_2 \succ_{s_2} p_1 \succ_{s_2} \emptyset$. The project preferences are $s_2 \succ_{p_1} s_1 \succ_{p_1} \emptyset$ and

[5] This example is identical to the example [16], which shows that a strongly stable matching may not exist with regional quotas.

$s_1 \succ_{p_2} s_2 \succ_{p_2} \emptyset$. By symmetry, we can assume the resource is allocated to p_1 w.l.o.g. For fairness, s_2 must be allocated to p_1. Then s_1 possibly claims an empty seat in p_2 since moving her to p_2 is possible by allocating the resource to p_2. □

Given this impossibility theorem, we introduce weaker conditions on efficiency.

Definition 12 *(Weak nonwastefulness).* For feasible matching (Y, μ), student s strongly claims an empty seat if $Y(s) = \emptyset$, and s claims an empty seat in project p with μ. A feasible matching is weakly nonwasteful if no student strongly claims an empty seat.

In the setting of Example 1, feasible matching $(\hat{Y}, \hat{\mu})$ in Fig. 1 is weakly nonwasteful because s_3 cannot strongly claim an empty seat. Although $\hat{Y}(s_3) = \emptyset$, she cannot be assigned to any project with current allocation $\hat{\mu}$.

Definition 13 *(Very weak nonwastefulness).* For feasible matching (Y, μ), student s *very strongly claims an empty seat* if $Y(s) = \emptyset$, and $\forall \mu'$, such that (Y, μ') is feasible, $\exists p$ in which s claims an empty seat with μ'. A feasible matching is very weakly nonwasteful if no student very strongly claims an empty seat.

In other words, student s very strongly claims an empty seat if she is currently unassigned, and under *any* feasible resource allocation μ', project p exists such that s claims an empty seat in p with μ'. Note that p can be different for each μ'.

Consider matching $Y = \{(s_1, p_1), (s_2, p_1), (s_3, \emptyset), (s_4, \emptyset)\}$ in the setting of Example 1. Then s_3 very strongly claims an empty seat. Here $Y(s_3) = \emptyset$. For any allocation with which Y is feasible, r_1 must be allocated to p_1. When r_2 is allocated to p_3, s_3 claims an empty seat in p_3. When r_2 is allocated to p_4, s_3 claims an empty seat in p_4.

If student s very strongly claims an empty seat, she is currently unassigned and claims an empty seat in project p with current allocation μ, and thus she also strongly claims an empty seat. If she claims an empty seat in p under the current assignment, she also possibly claims an empty seat in p. Thus, nonwastefulness implies weak nonwastefulness, and weak nonwastefulness implies very weak nonwastefulness.

To define another concept called *resource efficiency*, we first define *unanimous preferences*.

Definition 14 *(Unanimous preference).* Students unanimously prefer p over p' if for every $s \in S$, $(s, p) \in X$ and $p \succ_s p'$ hold.

This condition means that project p accepts all students and all students prefer p over p'. If students unanimously prefer p over p', allocating any resource (which is compatible with both p and p') to p' is *inefficient* in terms of students' welfare. The following formalizes this intuition.

Definition 15 *(Resource efficiency).* Resource allocation μ is resource efficient if no resource r, such that $p, p' \in T_r$ and students unanimously prefer p over p', is allocated to p'. A mechanism is resource efficient if it always returns a resource efficient allocation.

Pareto efficiency implies nonwastefulness. The following theorem shows that Pareto efficiency also implies resource efficiency.

Theorem 2. *If feasible matching (Y, μ) is Pareto efficient, then allocation μ' exists such that (Y, μ') is feasible and μ' is resource efficient.*

Proof. For the sake of contradiction, assume Y is Pareto efficient, and all students unanimously prefer p over p', but for any μ such that (Y, μ) is feasible, resource r exists such that $p, p' \in T_r$ is allocated to p'. Consider μ' obtained from μ, such that r is re-assigned to p. If (Y, μ') is feasible, we repeat the same procedure. (Y, μ') eventually becomes infeasible (otherwise, we obtain resource efficient μ', which contradicts our assumption). Since students unanimously prefer p over p', any student assigned to p' is acceptable to p and prefers p over p'. Consider another matching Y', in which some students are moved from p' to p such that (Y', μ') becomes feasible. Then the moved students prefer Y' over Y (and the other students are indifferent). This contradicts our assumption that Y is Pareto efficient. □

Now we are ready to introduce another impossibility theorem.

Theorem 3. *No mechanism exists that is fair, very weakly nonwasteful, resource efficient, and strategyproof.*

Proof. Consider the following situation: three students, s_1, s_2, s_3, three projects, p_1, p_2, p_3, one resource, r with $q_r = 2$, and $T_r = \{p_1, p_2, p_3\}$. The following are their preferences:

$$s_1 : p_2 \succ p_3 \succ p_1 \succ \emptyset, \quad p_1 : s_1 \succ s_2 \succ s_3 \succ \emptyset,$$
$$s_2 : p_3 \succ p_1 \succ p_2 \succ \emptyset, \quad p_2 : s_2 \succ s_3 \succ s_1 \succ \emptyset,$$
$$s_3 : p_1 \succ p_2 \succ p_3 \succ \emptyset, \quad p_3 : s_3 \succ s_1 \succ s_2 \succ \emptyset.$$

Recall that since all the resources must be distributed, resource r must be allocated to a project. From very weak nonwastefulness and fairness, the following are the possible matchings: allocating s_1 and s_2 to p_1, allocating s_2 and s_3 to p_2, or allocating s_3 and s_1 to p_3. From the symmetry, we can assume r is allocated to p_1 and s_1 and s_2 are assigned to p_1 w.l.o.g. Next, we examine the case where the preference of s_3 is changed to $p_3 \succ p_1 \succ p_2 \succ \emptyset$. From resource efficiency, r cannot be allocated to p_1 since all students prefer p_3 over p_1. If r is allocated to p_2 (or p_3), then from fairness and very weak nonwastefulness, s_3 must be assigned to p_2 (or p_3). This violates strategyproofness since s_3 is not assigned to any project in the original situation. □

4. Strategyproof mechanisms

4.1. Existing mechanisms

An SPR belongs to a general class of problems where distributional constraints satisfy a condition called *heredity* [2]. Heredity means that if matching Y is feasible (to be precise, if allocation μ exists such that (Y, μ) is feasible), then any of its subsets $Y' \subset Y$ is also feasible with some allocation μ'. SPR clearly satisfies this property: if (Y, μ) is feasible, for any $Y' \subset Y$, (Y, μ) is feasible. Three general strategyproof mechanisms exist in this context [2]. Since an SPR satisfies heredity, the properties of these mechanisms are automatically inherited to our model. Kamada and Kojima [48] also consider heredity constraints. They introduce a weaker stability concept called weak stability and show that a weakly stable matching always exists. Furthermore, Aziz et al. [34] introduce another stability concept called cut-off stability, which is stronger than weak stability, and show a cut-off stable matching always exists. These two stability concepts include fairness. Recently, Cho et al. [49] show that no weakly stable and strategyproof mechanism exists in general.

Before describing these mechanisms, we introduce a computational problem that needs to be solved within these mechanisms.

Definition 16 *(*Feasibility*).* For a given SPR instance and matching Y, does allocation μ exist such that (Y, μ) is feasible?

We settle its computational complexity by reduction from a partition problem, which is NP-complete [50].

Definition 17 *(*3DM*).* M, W, and D (men, women, and dogs) are disjoint sets of cardinality n. For a given set $\mathcal{F} \subseteq M \times W \times D$, does perfect matching F (i.e., a set $F \subseteq \mathcal{F}$ of disjoint families of cardinality n) exist?

Theorem 4. Feasibility *is NP-complete. This is true even if the capacity q_r is either 1 or 2 for all $r \in R$.*

Proof. For *yes* instances, whether (Y, μ) is feasible can be verified in polynomial time when μ is given as a certificate. Hence, this problem belongs to class NP.

We show that any instance of 3DM can be reduced to an SPR instance. The projects, resources, and assignment Y are created as follows.

- For each pair of a man and a woman $(m, w) \in M \times W$, we create project $p_{m,w}$.
- For each man $m \in M$, we create project p_m.
- For each woman $w \in W$, we create project p_w.
- In assignment Y, two students are assigned to each $p_{m,w}$, and one student is assigned to each p_m and each p_w.
- For each dog $d \in D$, we create resource r_d with $q_{r_d} = 2$. $T_r = \{p_{m,w} \mid (m, w, d) \in \mathcal{F}\}$, i.e., this resource can be assigned to $p_{m,w}$ when $(m, w, d) \in \mathcal{F}$.
- For each pair of a man and a woman $(m, w) \in M \times W$, we create two resources $r_{m,w}$ and $r_{w,m}$, with $q_{r_{m,w}} = q_{r_{w,m}} = 1$, $T_{r_{m,w}} = \{p_{m,w}, p_m\}$, and $T_{r_{w,m}} = \{p_{m,w}, p_w\}$.

Here, the total number of students and the total capacity of all resources are $2n(n+1)$. Thus, (Y, μ) is feasible only when for each project p, the total capacity of allocated resources is exactly equal to the number of assigned students.

Assume perfect matching F exists in the original 3DM. We can determine the feasible resource allocation μ as follows. For each project $p_{m,w}$, either (i) a unique element $(m, w, d) \in F$ exists, or (ii) for all $d \in D$, $(m, w, d) \notin F$ holds. For case (i), r_d is assigned to $p_{m,w}$. For case (ii), $r_{m,w}$ and $r_{w,m}$ are assigned to $p_{m,w}$. For each project p_m, a unique element $(m, w, d) \in F$ exists. $r_{m,w}$ is assigned to p_m. For each project p_w, a unique element $(m, w, d) \in F$ exists. $r_{w,m}$ is assigned to p_w. It is clear that (Y, μ) is feasible.

Next, assume resource allocation μ exists such that (Y, μ) is feasible. Each r_d must be allocated to $p_{m,w}$, such that $(m, w, d) \in \mathcal{F}$. Let us choose $F = \{(m, w, d) \mid d \in D, r_d \text{ is allocated to } p_{m,w}\}$. Clearly, $|F| = n$ holds. We show that F is a perfect matching, i.e., if r_{d_1} is assigned to p_{m_1,w_1} and r_{d_2} is assigned to p_{m_2,w_2}, $m_1 \neq m_2$ and $w_1 \neq w_2$ hold.

By way of contradiction, assume $m_1 = m_2 = m$ holds. Since r_{d_1} is assigned to p_{m,w_1}, r_{m,w_1} must be assigned to p_m (otherwise, p_{m,w_1} is assigned excessive resources, which means that another resource is not assigned enough resources). Similarly, since r_{d_2} is assigned to p_{m,w_2}, r_{m,w_2} must be assigned to p_m. However, since the demand of p_m is one, it is assigned excessive resources to satisfy its demand. Thus, another project exists that is not assigned enough resources. This contradicts our assumption that μ is a feasible allocation.

Similarly, by way of contradiction, assume $w_1 = w_2 = w$ holds. Since r_{d_1} is assigned to $p_{m_1,w}$, r_{w,m_1} must be assigned to p_w. Also, since r_{d_2} is assigned to $p_{m_2,w}$, r_{w,m_2} must be assigned to p_w. However, since the demand of p_w is one, it is assigned excessive resources to satisfy its demand. Thus, another project exists that is not assigned enough resources. This contradicts our assumption that μ is a feasible allocation.

Since 3DM is NP-complete, Feasibility is also NP-complete. □

To verify feasibility, we need to solve a Mixed Integer Programming (MIP) instance. Note that this impossibility crucially depends on the fact that each resource is indivisible. When resources are divisible, i.e., we can assume that $q_r = 1$ for all r, Feasibility is no longer NP-complete; we can solve it using a max-flow algorithm [51]. Furthermore, if T_R has a laminar structure (for any T_r and $T_{r'}$, one of the following three conditions hold (i) $T_r \subseteq T_{r'}$, (ii) $T_r \supsetneq T_{r'}$, or (iii) $T_r \cap T_{r'} = \emptyset$), we can translate such distributional constraints into hierarchical regional quotas which form an M$^\natural$-convex set [14]. Then, the generalized DA mechanism [14] is strategyproof and fair.

In the following, we describe these mechanisms adopted to an SPR one by one. First, Serial Dictatorship mechanism (SD) uses a serial order among the students. Although the order can be arbitrary, it must be determined independently of student preferences to guarantee strategyproofness. W.l.o.g., we assume this order is $s_1, s_2, \ldots$.

Conceptually, SD can be described as follows. Let $\boldsymbol{Y}$ denote all the possible matchings, each of which can be feasible with some allocation. The first student, s_1, chooses subset $\boldsymbol{Y}_1 \subseteq \boldsymbol{Y}$, such that she equally prefers any matching in $\boldsymbol{Y}_1$ and strictly prefers any matching in $\boldsymbol{Y}_1$ over any matching in $\boldsymbol{Y} \setminus \boldsymbol{Y}_1$. In other words, she chooses her most preferred matchings in $\boldsymbol{Y}$. Since she is concerned with the project to which she is assigned and has no interest in the assignments of other students, her most preferred matching is not unique, and her choice is a subset of $\boldsymbol{Y}$. In the setting of Example 1, s_1 will choose $\boldsymbol{Y}_1 = \{Y \in \boldsymbol{Y} \mid (s_1, p_1) \in Y\}$, i.e., all the elements in $\boldsymbol{Y}$ such that s_1 is allocated to p_1. Then the next student s_2 chooses $\boldsymbol{Y}_2 \subseteq \boldsymbol{Y}_1$ in a similar way; she chooses her most preferred matchings within $\boldsymbol{Y}_1$, and so forth. In the setting of Example 1, s_2 will choose $\boldsymbol{Y}_2 = \{Y \in \boldsymbol{Y}_1 \mid (s_2, p_1) \in Y\}$, i.e., all the elements in $\boldsymbol{Y}_1$ such that s_2 is allocated to p_1. SD is clearly strategyproof since each student can choose her most preferred matchings from the exogenously determined possibilities. SD is also Pareto efficient for the following reason. Clearly, we cannot improve the assignment of s_1. Nor can we improve s_2 without hurting s_1, and so forth. Thus, it is impossible to improve the assignment of one student without hurting other students. Since SD is Pareto efficient, it is also nonwasteful.

The following is the formal definition of SD for an SPR:

Mechanism 4.1 *(Serial Dictatorship (SD)).*

Step 1: $Y \leftarrow \emptyset$. $k \leftarrow 1$.
Step 2: If $k > |S|$, return Y. Otherwise, choose $(s_k, p) \in X$, where p is her most preferred, acceptable project such that $Y \cup \{(s_k, p)\}$ is feasible with some allocation μ'. $Y \leftarrow Y \cup \{(s_k, p)\}$ (if no such p exists, s_k is not assigned to any project).
Step 3: $k \leftarrow k + 1$. Go to Step 2.

Unfortunately, SD is excessively *unfair*; many students could have justified envy in SD since it completely ignores project preferences.

The next mechanism is Artificial Caps Deferred Acceptance (ACDA), which is based on the well-known Deferred Acceptance (DA) [5]. In DA, each student first applies to her most preferred project. Then each project provisionally accepts students up to its capacity limit based on its preference and rejects the rest of them. A rejected student applies to her second choice. Each project provisionally accepts students who have applied without distinguishing between newly applied and already provisionally accepted students, and so forth. To apply DA, the maximum quota (i.e., capacity limit) of each project must be predetermined. In ACDA, we artificially determine maximum quotas. More specifically, we choose an arbitrary allocation μ independently of student preferences and decide the maximum quotas based on it.

The detailed procedure of ACDA for an SPR is given as follows:

Mechanism 4.2 *(Artificial Caps Deferred Acceptance (ACDA)).*

Step 1: Choose μ independently of $\succ_S$.
Step 2: Run the standard DA, assuming the maximum quota of each project p is $\sum_{r \in \mu^{-1}(p)} q_r$ and obtain matching Y.
Step 3: Return (Y, μ).

Although ACDA obtains a fair matching in polynomial-time, it can be very inefficient; many students would possibly claim an empty seat since μ is chosen independently of their preferences.

Table 1
Mechanism Properties.

	SP	Fairness	PE	NW	Weak NW	RE
SD [2]	yes	no	yes	yes	yes	yes
ADA [2]	yes	no	no	yes	yes	yes
SDA-V	yes	no	no	no	yes	yes
SDA-V*	yes	no	no	no	yes	yes
SDA-S	yes	no	no	no	yes	yes
ACDA [2]	yes	yes	no	no	yes	no

The third mechanism is Adaptive Deferred Acceptance (ADA). Like SD, ADA uses a serial order among students. The order can be arbitrary, but it must be determined independently of student preferences to guarantee strategyproofness. W.l.o.g., we assume this order is $s_1, s_2, \ldots$. As well as ACDA, ADA requires maximum quota q_p for each project p. If no maximum quota is given, i.e., if we assume $q_p = \infty$ for each $p \in P$, ADA obtains identical matching as SD. To apply ADA to an SPR, we choose q_p as $\sum_{r | T_r \ni p} q_r$, which is the largest capacity when all the compatible resources are allocated to it. During the execution of ADA, we say project p is forbidden under (partial) matching Y if the following conditions hold: (i) $Y \cup \{(s, p)\}$ is infeasible with any allocation, and (ii) $|Y(p)| < q_p$. In other words, project p is forbidden if p cannot accept another student due to resource contention among projects, even though it does not reach its maximum quota. Formally, ADA for an SPR is defined as follows:

Mechanism 4.3 *(Adaptive Deferred Acceptance (ADA)).* We initially assume that no project is forbidden. Let $L \leftarrow (s_1, s_2, \ldots)$, $q_p^1 \leftarrow q_p$ for each $p \in P$, $Y \leftarrow \emptyset$. Proceed to **Stage** 1.

Stage k: Proceed to **Round** 1.
Round t: Select t students from the top of L. Let Y' denote the matching obtained by DA for the selected students under $(q_p^k)_{p \in P}$.
(i) If all students in L are already selected, then output $Y \cup Y'$ and terminate the mechanism.
(ii) If no project p_i exists that is forbidden, then proceed to **Round** $t+1$.
(iii) Otherwise, $Y \leftarrow Y \cup Y'$. Remove t students from the top of L. For each project p that is forbidden, set q_p^{k+1} to 0. For each $p \in P$, which is not forbidden, set q_p^{k+1} to $q_p^k - |Y'(p)|$. Proceed to **Stage** $k+1$.

We can assume ADA combines SD and DA, in which student groups are sequentially allocated as SD, but within each group, students compete with each other by DA. We show how ADA works in the setting of Example 1 The maximum quotas of the projects are determined as $(2, 2, 1, 1)$. First, in Round 1 of Stage 1, running DA and s_1 is assigned to p_1. Then project p_2 is forbidden. Although its maximum quota is two and no student is currently assigned to it, we cannot allocate another student to it since r_1 is already taken by p_1 to accommodate s_1. Thus, the assignment (s_1, p_1) is fixed. The maximum quotas are reset to $(1, 0, 1, 1)$. Then in Round 1 of Stage 2, s_2 is assigned to p_1 by DA. No project is forbidden (note that p_1 has already reached its maximum quota and it is not forbidden). In Round 2 of Stage 2, s_2 is assigned to p_3, and s_3 is assigned to p_1 using DA. Then project p_4 is forbidden. Although its maximum quota is one and no student is currently assigned to it, we cannot allocate another student to it since r_2 is taken by p_3 to accommodate s_2. Thus, the assignments (s_2, p_3) and (s_3, p_1) are fixed. The new maximum quotas become $(0, 0, 0, 0)$. Thus, no more students can be assigned. ADA terminates and returns $\{(s_1, p_1), (s_2, p_3), (s_3, p_1), (s_4, \emptyset)\}$. ADA is nonwasteful because project p is forbidden only when by allocating another student to p, there is no way to make the current matching feasible. However, it is computationally as expensive as SD since we need to solve Feasibility for checking whether a project is forbidden.

Using these mechanisms, we can show that resource efficiency and very weak nonwastefulness are independent properties. For example, ACDA satisfies weak nonwastefulness (thus it also satisfies very weak nonwastefulness). In ACDA, with resource allocation μ (which is used for determining maximum quotas), no student claims an empty seat, although it does not satisfy resource efficiency, i.e., it may allocate a resource to a unanimously less preferred project. Next, assume a mechanism assigns no student to any project, and no resource is allocated to any project p' if students unanimously prefer another project p. It is trivially strategyproof and satisfies resource efficiency. However, this mechanism does not always satisfy very weak nonwastefulness. Table 1 summarizes the properties of these mechanisms, where SP stands for strategyproofness, PE for Pareto efficiency, NW for nonwastefulness, and RE for resource efficiency.

4.2. New mechanisms

ACDA is excessively inefficient, and SD and ADA are too unfair (many students have justified envy). Moreover, Theorem 3 shows that fairness cannot be achieved without significantly sacrificing efficiency. To establish a good balance between fairness and efficiency in SPR problems, we introduce a new class of mechanisms called *Sample and Deferred Acceptance* (SDA).

SDA is outlined as follows.

Mechanism 4.4 *(Sample and Deferred Acceptance (SDA)).*

Step 1: Select $S' \subseteq S$ independently of $\succ_S$, which we call sampled students, and $S' \neq \emptyset$. We call $S \setminus S'$ the regular students. Then run SD and find (partial) matching $Y_{S'}$ for S'.
Step 2: Allocate $R' \subseteq R$ to projects such that $Y_{S'}$ is feasible and R' is minimal: no $R'' \subsetneq R'$ makes $Y_{S'}$ feasible. Then allocate $R \setminus R'$ based on the preferences of S'.
Step 3: Run DA for $S \setminus S'$. The capacity of p is $q_\mu(p) - |Y_{S'}(p)|$, where μ is the resource allocation determined in Step 2.

The entire mechanism is carefully designed to guarantee strategyproofness. We adopt a popular technique used in auction domains to ensure strategyproofness, where a mechanism has some parameters and their selection affects participants' welfare [52,53]. The idea is first dividing students/participants into two groups, then extracting the information from one group, and lastly setting the parameters of the mechanism applied to the other group based on the obtained information. To the best of our knowledge, applying this idea in two-sided matching to develop a strategyproof mechanism is novel.

By using different methods to determine resources in Step 2, we can create a variety of sample-based mechanisms that can strike a good balance between fairness and efficiency by slightly sacrificing fairness to improve efficiency. We introduce a first strategyproof sample-based mechanism, which we call Sample and Deferred Acceptance with Voting (SDA-V). In Step 2 of Mechanism 4.4, SDA-V determines resource allocation μ based on the voting among sampled students. More specifically, each sampled student (hypothetically) votes for all projects based on $\succ_s$, where each project obtains a Borda score, i.e., the top project obtains $|P|$, the second project obtains $|P|-1$, and so on. Then for each project, the sum of these scores is calculated. Finally, each resource r is allocated to the project that obtains the highest total score within T_r. The details of this voting procedure do not affect SDA-V's theoretical properties, e.g., whether a student can vote for a project to which she is unacceptable, or how ties are broken. Thus, they can be arbitrarily determined.

We show how SDA-V works with the following example.

Example 2. There are five students, s_1, s_2, s_3, s_4, s_5, four projects, p_1, p_2, p_3, p_4, and four resources, r_1, r_2, r_3, r_4, where $T_{r_1} = \{p_1, p_2\}$, $T_{r_2} = \{p_1, p_2, p_3\}$, $T_{r_3} = \{p_3, p_4\}$, $T_{r_4} = \{p_2, p_3, p_4\}$, and $q_{r_1} = q_{r_3} = q_{r_4} = 1$, and $q_{r_2} = 2$. The following are their preferences:

$$\begin{array}{ll}
s_1 : p_1 \succ p_2 \succ p_3 \succ p_4 \succ \emptyset, & p_1 : s_5 \succ s_4 \succ s_1 \succ s_3 \succ s_2 \succ \emptyset,\\
s_2 : p_4 \succ p_2 \succ p_3 \succ p_1 \succ \emptyset, & p_2 : s_5 \succ s_4 \succ s_3 \succ s_2 \succ s_1 \succ \emptyset,\\
s_3 : p_1 \succ p_2 \succ p_3 \succ p_4 \succ \emptyset, & p_3 : s_1 \succ s_2 \succ s_3 \succ s_4 \succ s_5 \succ \emptyset,\\
s_4 : p_4 \succ p_2 \succ p_3 \succ p_1 \succ \emptyset, & p_4 : s_1 \succ s_2 \succ s_3 \succ s_4 \succ s_5 \succ \emptyset.\\
s_5 : p_3 \succ p_2 \succ p_1 \succ p_4 \succ \emptyset, &
\end{array}$$

In Step 1, assume $S' = \{s_1, s_2\}$. In SD, s_1, s_2 are matched to their first-choice projects, p_1, p_4 respectively. In Step 2, the minimal allocation to make $\{(s_1, p_1), (s_2, p_4)\}$ feasible is allocating r_1 to p_1 and r_3 to p_4. Thus, $R' = \{r_1, r_3\}$. Next the allocation of $R \setminus R' = \{r_2, r_4\}$ is determined by the preferences of s_1 and s_2. The Borda scores of the four projects are $(5, 6, 4, 5)$, so r_2 and r_4 are allocated to p_2. In Step 3, since s_1 and s_2 are fixed, the remaining capacities of p_1, p_2, p_3, and p_4 are $0, 3, 0$, and 0. $S \setminus S' = \{s_3, s_4, s_5\}$ are matched by DA. Thus, s_3, s_4, and s_5 are matched to p_2. The result is weakly nonwasteful, but it is not fair since s_3 has justified envy toward s_1. Nonwastefulness is not satisfied either since s_3 possibly claims an empty seat in p_1.

SDA-V uses a simple voting scheme for deciding resource allocation. However, if students are divided into several groups in terms of their preferences, we might obtain a sub-optimal result, as illustrated by the following example.

Example 3. We use the same instance as Example 2 except that the students are divided into two types, i.e., types 1 and 2; there are three type 1 students and two type 2 students. The preference of each type 1 student is $p_1 \succ_s p_2 \succ_s p_3 \succ_s p_4 \succ_s \emptyset$, and the preference of each type 2 student is $p_4 \succ_s p_2 \succ_s p_3 \succ_s p_1 \succ_s \emptyset$. Assume that one type 1 student and one type 2 students are sampled. Since the Borda scores of the four projects are $(5, 6, 4, 5)$, SDA-V allocates r_2 and r_4 to p_2. However, if we allocate r_2 to p_1 and r_4 to p_4 instead, all the students in these two groups can be assigned to their first-choice projects.

As illustrated by Example 3, the Borda count is somewhat too coarse-grained to determine an appropriate resource allocation. More specifically, for each resource r, we choose a single winner among projects T_r in SDA-V. By using the Borda count, we can choose a broadly acceptable project among candidates T_r. Example 3 shows that when multiple resources are available and students are divided into several groups, choosing a single winner can be sub-optimal; resources should be divided proportionally to multiple winners.[6]

[6] Actually, SDA-V's performance in terms of students' welfare can be as bad as ACDA when students are divided into different types in Section 5.

By using a different voting scheme, which selects multiple winners and allocates resources proportionally to them, we can allocate resources such that all groups are satisfied to a certain extent. To be more precise, we can use the following voting scheme:

- For each resource r, sampled students vote for T_r. Each sampled student casts a single vote for her favorite project within T_r.
- Let $w(p)$ denote the number of votes obtained by each project $p \in T_r$. Each project is allocated to r with probability $w(p)/\sum_{p' \in T_r} w(p')$.

We call SDA with this modified voting scheme SDA-V*. As shown in Section 5, SDA-V* outperforms SDA-V when students are divided into several groups.

We develop yet another way for deciding resource allocation, which does not rely on any voting scheme. Instead of aggregating the preference by voting, we create copies of sampled students and run a simulation for them to decide the resource allocation. More specifically, we (imaginarily) perform SD mechanism for these copied students and obtain a resource allocation. We call this mechanism Sample and Deferred Acceptance with Simulation (SDA-S). Compared to SDA-V or SDA-V*, SDA-S fully utilizes the preference of each sampled student, i.e., not only her first-choice project, but also her second-choice, third-choice, and so on.

In SDA-S, the detailed procedure of Step 2 in Mechanism 4.4 is given as follows:

1. Make $n - |S'|$ copies of sampled students (i.e., for each sampled student, we create either $\lfloor \frac{n-|S'|}{|S'|} \rfloor$ or $\lceil \frac{n-|S'|}{|S'|} \rceil$ copies).
2. Run SD for these copies to determine the resource allocation. First, we create copied students and the serial order of them is determined as follows. Assume sampled students are $s_1, s_2, \ldots, s_{|S'|}$. Let d_j^i denote the j-th copy of student s_i. Then the serial order is $d_1^1, d_1^2, \ldots, d_1^{|S'|}, d_2^1, d_2^2, \ldots, d_2^{|S'|}, d_3^1, \ldots$, i.e., using a round-robin order among the sampled students to determine the serial order of the copied students. Second, we run SD for these copied students to find a (imaginary) matching that is feasible with the remaining resources. The resources allocated by the imaginary matching is used for DA in Step 3 of Mechanism 4.4.

We illustrate how SDA-S works. Assume the setting in Example 2. In Step 1, assume $S' = \{s_1, s_2\}$. In SD, s_1, s_2 are matched to their first-choice projects, p_1, p_4 respectively. In Step 2, the minimal allocation to make $\{(s_1, p_1), (s_2, p_4)\}$ feasible is allocating r_1 to p_1 and r_3 to p_4. Thus, $R' = \{r_1, r_3\}$. Next the allocation of $R \setminus R' = \{r_2, r_4\}$ is determined by running SD for copies. We create two copies of s_1 (i.e., d_1^1, d_2^1) and one copy of s_2 (i.e., d_1^2). In SD, d_1^1, d_2^1 are assigned to p_1, and d_1^2 is assigned to p_4. Then r_2 is allocated to p_1, and r_4 is allocated to p_4 to make the imaginary matching feasible. In Step 3, since s_1 and s_2 are fixed, the remaining capacities of p_1, p_2, p_3, and p_4 are $2, 0, 0$, and 1. $S \setminus S' = \{s_3, s_4, s_5\}$ are matched by DA. Thus, s_3, s_5 are matched to p_1, and s_4 is matched to p_4.

4.3. Characteristics of new mechanisms

We describe that the family of SDA mechanisms satisfy several fundamental desiderata.

Theorem 5. *Any SDA instance is strategyproof, weakly nonwasteful, fair among students in $S \setminus S'$, and no sampled student has justified envy toward another regular student.*

Proof. First, the sampled students S' are chosen independently of students' preferences (e.g., by random sampling). Thus, students cannot affect who will be selected in S'. Then, for each student in S', she has no incentive to misreport since her assignment is determined by SD, and the resource allocation for students $S \setminus S'$ (which is determined using her preference) is irrelevant to her. Next, for each student in $S \setminus S'$, the capacity of each project is determined independently of her preference. Also, since DA is strategyproof [54,55], she has no incentive to misreport her preference.

Assume student s is matched to p (which can be $\emptyset$). She applied to project p', which outranks p, and was rejected. If $s \in S'$, then no feasible allocation μ' exists such that s can be assigned to p'. If $s \in S \setminus S'$, s cannot be assigned to p' with current allocation μ. Hence, any SDA instance is weakly nonwasteful.

Regarding fairness, since DA is fair [5], no regular student has justified envy toward another regular student. If sampled student $s \in S'$ is rejected by p, then no more students can be assigned to p. Thus, s never has justified envy toward a regular student who is assigned after s. $\square$

Theorem 6. *Any SDA instance is weakly Pareto efficient.*

Proof. Assume s is the first sampled student. In other words, s is ordered first in SD. She is eventually assigned to her most preferable project p among all projects such that at least one resource r can be allocated with respect to T_r. If another project p' exists such that $p' \succ_s p$ holds, p' has no compatible resource. Thus, it is impossible to assign s to p', and we cannot strictly improve s's allocation. Hence, no feasible matching exists that strongly Pareto dominates the matching obtained by SDA. □

Theorem 7. *Any SDA instance is weakly group strategyproof.*

Proof. Since SD is group strategyproof [56,57], a sampled student cannot benefit by joining a coalition of sampled students. Furthermore, regular students never affect the assignment of sampled students. Thus, a sampled student cannot benefit by joining a coalition of sampled and regular students.

Since DA is weakly group strategyproof [58] and project quotas are exogenously given by the preferences of the sampled students, no coalition of regular students can collude to misreport their preferences. Hence, no group of students has an incentive to collude and weak group strategyproofness holds for sample-based mechanisms. □

Note that a stronger requirement exists against a deviation of a group called strong group strategyproofness. A mechanism is strongly group strategyproof if no group of students can collude to misreport their preferences in a way that makes at least one member strictly better off and all members weakly better off. In contrast to weak group strategyproofness, a colluding group can include a member whose assignment does not change. Pápai [57] shows a characterization of strong group strategyproofness: a mechanism is strongly group strategyproof if and only if it is strategyproof and *non-bossy*. A mechanism is non-bossy if a student cannot change the assignment of another student without changing her own assignment.

SDA is clearly bossy; a sampled student can change the assignment of a regular student without changing her assignment. As a result, SDA is not strongly group strategyproof. More specifically, a coalition can be formed such that a subset of students benefits while the assignments of other students do not change, e.g., a sampled student can manipulate her vote to favor some regular students even though she does not benefit from this action. Since weak group strategyproofness requires that *all members* must benefit, the existence of such a coalition does not contradict the fact that any SDA instance is weakly group strategyproof. Note that DA itself is bossy (thus it is not strongly group strategyproof). Assume three students, s_1, s_2, s_3, and three projects, p_1, p_2, p_3, such that each capacity is one. Student s_1 applies to project p_1, which makes s_2 to be rejected by p_1. Student s_2 then applies to p_2, which makes s_3 to be rejected by p_2. Next, s_3 applies to p_1, which makes s_1 to be rejected by p_1. Finally, s_1 applies to p_3 and all students are accepted. If s_1 first applies to p_3, her assignment will not change although s_2 will be accepted to p_1 and s_3 will be accepted to p_2. Thus, s_1 can change the assignment of another student without changing her assignment in DA. These students are also able to collude and misreport their preferences such that s_2 and s_3 are strictly better off making s_1's assignment unchanged.

Whether an SDA instance satisfies resource efficiency depends on how the resources are allocated based on the preferences of the sampled students. We show that the procedures for allocating resources in SDA-V, SDA-V*, and SDA-S are carefully designed such that they satisfy resource efficiency.

Theorem 8. *SDA-V, SDA-V*, and SDA-S are resource efficient.*

Proof. Assume students unanimously prefer p over p'. When assigning sampled students S', students apply to p before applying to p'. Thus, we do not require that any resource r is allocated to p' such that $p, p' \in T_r$ holds. This property holds when assigning copied students by SD in SDA-S. Furthermore, r' never wins in the voting procedure in SDA-V and SDA-V*. Thus, these mechanisms satisfy resource efficiency. □

Next, we show that SDA mechanisms satisfy additional properties (i.e., Pareto efficiency and fairness) in special cases where project or student preferences are identical.

Theorem 9. *Any instance of SDA mechanisms is fair if all the projects have an identical preference and the sampled students are selected based on it.*

Proof. From Theorem 5, if s has justified envy toward s', then there are two cases: (i) s is a regular student and s' is a sampled student, or (ii) both s and s' are sampled students. Assume student s, who is assigned to project p, has justified envy toward another student s', who is assigned to p' (i.e., $s \succ_{p'} s'$ and $p' \succ_s p$ hold).

For case (i), $s' \succ_{p'} s$ must hold from the assumption that every project unanimously prefers a sampled student over a regular student, which contradicts our assumption that $s \succ_{p'} s'$ holds. For case (ii), s' must be assigned before s, which means every project unanimously prefers s' over s. This contradicts our assumption that $s \succ_{p'} s'$ holds. □

Theorem 10. *SDA-V, SDA-V*, and SDA-S are Pareto efficient if all student preferences are identical and each project assumes that all students are acceptable.*

Proof. W.l.o.g., assume the preference of each student is $p_1 \succ_s p_2 \succ_s \dots$. Since SD is Pareto efficient, it is impossible to assign sampled student s to a better project without disadvantaging another sampled student s' who was assigned before s.

For SDA-V and SDA-V*, resource allocation μ is determined by the votes of the sampled students, so any resource r in μ is allocated to p_i such that it has the smallest identifier in all the projects within T_r. Since all student preferences are identical and each project assumes that all students are acceptable, no better allocation exists that can improve the assignment of the regular students.

For SDA-S, resource allocation μ is determined by running SD on the copied students. We cannot further improve this imaginary matching without making a copied student worse off. Furthermore, all student preferences are identical, and each project assumes that all students are acceptable. Thus, allocation μ cannot be improved without sacrificing one regular student's welfare.

In DA, all regular students $S \setminus S'$ first apply to p_1. Assume a set of regular students S_1 is accepted to p_1 and the remaining students are rejected. By repeating a similar procedure, students in S_k are accepted to p_k. It is impossible to assign a student in S_k to a better project without affecting the students in S' or $S_{k'}$ (where $k' < k$). Thus, no matching Pareto dominates the matching obtained by any of these mechanisms. □

Any SDA instance needs to verify feasibility $O(|S' \times P|)$ times in Step 1. However, when $|S'|$ is small, such a feasibility problem is trivially *yes* in most cases, assuming projects are equipped with a reasonable amount of resources (the amount of resources might not be able to satisfy the demand of all students, but it should be sufficient for a small set of them). In addition to the sampled students, SDA-S also needs to solve $O((n - |S'|) \times |P|)$ times for the copied students. In total, SDA-S requires solving Feasibility $O(|n \times P|)$ times. Thus, in terms of computational cost, SDA-S and SD are equivalent while they are more expensive than SDA-V and SDA-V*. Although SDA-S and SD are computationally expensive, state-of-the-art MIP solvers, e.g., the Gurobi optimizer [59], can handle fairly large-scale feasibility problems.

5. Experiment

5.1. Setup

We consider two markets, I and II, which have different sizes. Market I has 200 students, 10 projects, and 20 resources, and Market II has 400 students, 20 projects, and 40 resources.[7] For each resource r, we randomly generate T_r such that each project p is included in T_r with probability $\gamma = 0.2$, 0.5, and 0.8. Student preferences are generated with the Mallows model [60–63]. In this model, student preference $\succ_s$ is drawn with probability $\Pr(\succ_s)$:

$$\Pr(\succ_s) = \frac{\exp(-\phi \cdot d(\succ_s, \succ_{\widehat{s}}))}{\sum_{\succ'_s} \exp(-\phi \cdot d(\succ'_s, \succ_{\widehat{s}}))}.$$

Here $\phi \in \mathbb{R}$ denotes a spread parameter, $\succ_{\widehat{s}}$ is a central preference, and $d(\succ_s, \succ_{\widehat{s}})$ represents the Kendall tau distance, which is the number of pairwise inversions between $\succ_s$ and $\succ_{\widehat{s}}$. In short, student preferences are distributed around a central preference with spread parameter ϕ. When $\phi = 0$, the Mallows model becomes identical to the uniform distribution (which is equivalent to the impartial culture [64,65] in our setting), and as ϕ increases, it quickly converges to the constant distribution that returns $\succ_{\widehat{s}}$.

Based on the Mallows model, we generated students' preferences for the following three cases in two markets:

1-type case: For Market I, we fix the central preference: $\succ_{\widehat{s_1}} = p_1 \succ p_2 \succ \ldots \succ p_9 \succ p_{10}$, and the preference of each student is distributed around $\succ_{\widehat{s_1}}$. For Market II, we also fix the central preference: $\succ_{\widehat{s'_1}} = p_1 \succ p_2 \succ \ldots \succ p_{19} \succ p_{20}$.

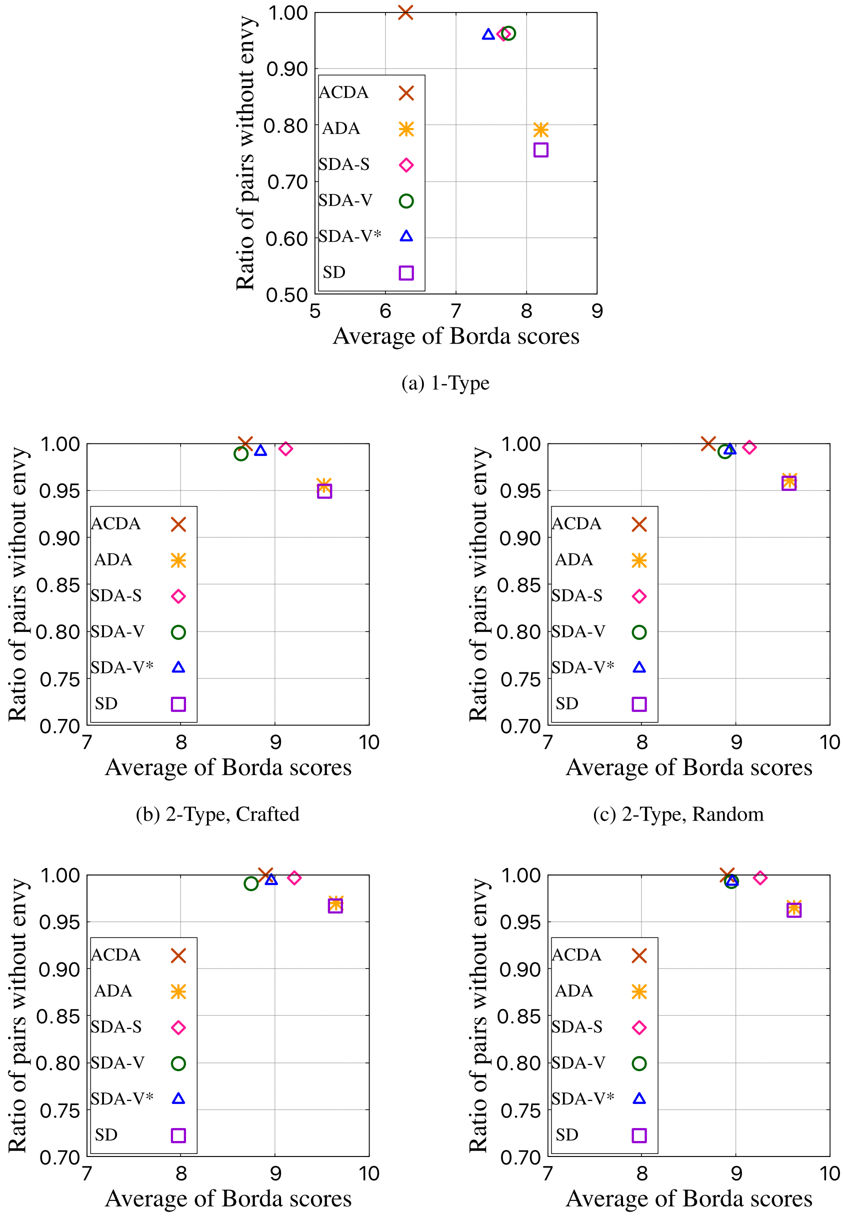
2-type case:

Crafted: For Market I, we first divide the projects into three groups: group $A = \{p_1, p_2, p_3\}$, group $B = \{p_4, p_5, p_6, p_7\}$, and group $C = \{p_8, p_9, p_{10}\}$. Then $\succ_{\widehat{s_1}}$ can be represented as ABC, in which group A is first, group B is second, and group C is third (for two projects p_i and p_j in the same group, the order is $p_i \succ_s p_j$ assuming $i < j$ holds). Next, we create another central preference represented as CBA: $\succ_{\widehat{s_2}} = p_8 \succ p_9 \succ p_{10} \succ p_4 \succ p_5 \succ p_6 \succ p_7 \succ p_1 \succ p_2 \succ p_3$. These preferences are crafted such that a project in group B is likely to be a winner in terms of the Borda count. Students S are divided into two groups S_1 and S_2, where $|S_1| = |S_2| = 100$. For each $s \in S_1$, her preference is distributed around $\succ_{\widehat{s_1}}$, and for each $s \in S_2$, her preference is distributed around $\succ_{\widehat{s_2}}$.

For Market II, we also divide the projects into three groups: group $A' = \{p_1, p_2, p_3 p_4, p_5, p_6\}$, group $B' = \{p_7, p_8, p_9, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}\}$, and group $C' = \{p_{15}, p_{16}, p_{17}, p_{18}, p_{19}, p_{20}\}$. Then $\succ_{\widehat{s'_1}}$ can be represented as $A'B'C'$. Next we create another central preference represented as $C'B'A'$: $\succ_{\widehat{s'_2}} = p_{15} \succ p_{16} \succ p_{17} \succ p_{18} \succ p_{19} \succ p_{20} \succ p_7 \succ p_8 \succ p_9 \succ p_{10} \succ p_{11} \succ p_{12} \succ p_{13} \succ p_{14} \succ p_1 \succ p_2 \succ p_3 \succ p_4 \succ p_5 \succ p_6$.

Random: We also examine the case where two central preferences are randomly chosen. For Market I, we use two randomly chosen central preferences: $\succ_{\widehat{s_3}} = p_5 \succ p_4 \succ p_3 \succ p_2 \succ p_9 \succ p_8 \succ p_1 \succ p_7 \succ p_{10} \succ p_6$, and $\succ_{\widehat{s_4}} = p_8 \succ p_6 \succ p_1 \succ p_2 \succ p_{10} \succ p_5 \succ p_7 \succ p_3 \succ p_9 \succ p_4$.

[7] Since SD, ADA, and SDA-S are computationally expensive, running experiments for larger markets is time-consuming, although SDA-V, SDA-V*, and ACDA can handle much larger markets.



(a) 1-Type

(b) 2-Type, Crafted

(c) 2-Type, Random

(d) 3-Type, Crafted

(e) 3-Type, Random

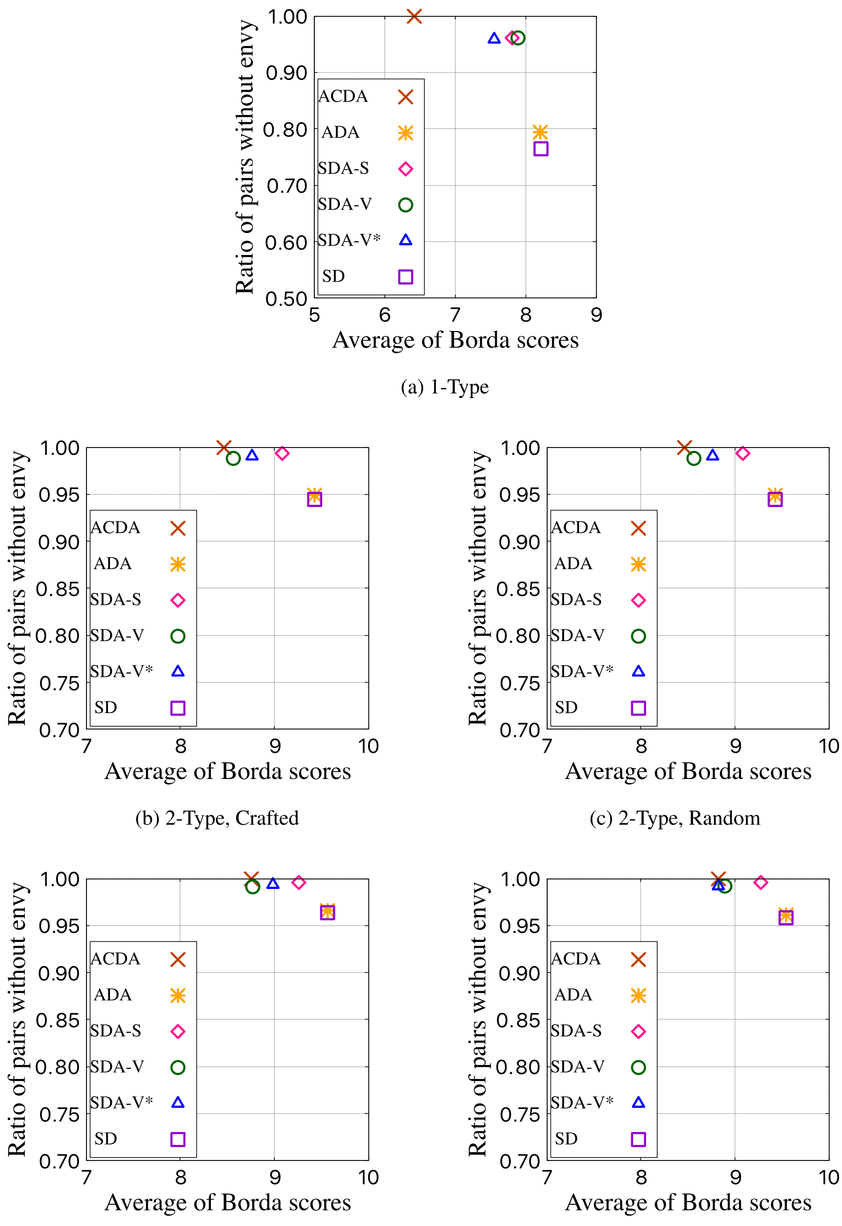
Fig. 2. Trade-off between efficiency and fairness ($\phi = 0.8$, $\rho = 0.1$, $\gamma = 0.2$, homogeneous resources, Market I). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

For Market II, we use two randomly chosen central preferences: $\succ_{\widehat{s'_3}} = p_{14} \succ p_{11} \succ p_{16} \succ p_7 \succ p_9 \succ p_8 \succ p_{10} \succ p_{17} \succ p_{18} \succ p_6 \succ p_1 \succ p_5 \succ p_{15} \succ p_2 \succ p_3 \succ p_{19} \succ p_4 \succ p_{13} \succ p_{20} \succ p_{12}$, and $\succ_{\widehat{s'_4}} = p_{12} \succ p_{16} \succ p_{14} \succ p_9 \succ p_{10} \succ p_2 \succ p_7 \succ p_1 \succ p_{17} \succ p_{20} \succ p_{15} \succ p_{11} \succ p_8 \succ p_3 \succ p_{19} \succ p_5 \succ p_{18} \succ p_{13} \succ p_6 \succ p_4$.

3-type case:

Crafted: For Market I, we first divide projects into four groups: group $A = \{p_1, p_2\}$, group $B = \{p_3, p_4, p_5, p_6\}$, group $C = \{p_7, p_8\}$, and group $D = \{p_9, p_{10}\}$. Thus, $\succ_{\widehat{s_1}}$ can be represented as $ABCD$. Next, we create additional two central preferences $\succ_{\widehat{s_5}}$ and $\succ_{\widehat{s_6}}$, which can be represented as $CBDA$ and $DBAC$, respectively. More precisely, we choose $\succ_{\widehat{s_5}} = p_7 \succ p_8 \succ p_3 \succ p_4 \succ p_5 \succ p_6 \succ p_9 \succ p_{10} \succ p_1 \succ p_2$, and $\succ_{\widehat{s_6}} = p_9 \succ p_{10} \succ p_3 \succ p_4 \succ p_5 \succ p_6 \succ p_1 \succ p_2 \succ p_7 \succ p_8$. These preferences are crafted such that a project in group B is likely to be a winner in terms of the Borda count. The population for each central preference is 66 or 67.

For Market II, we similarly divide projects into four groups: group $A' = \{p_1, p_2, p_3, p_4, p_5\}$, group $B' = \{p_6, p_7, p_8, p_9, p_{10}\}$, group $C' = \{p_{11}, p_{12}, p_{13}, p_{14}, p_{15}\}$, and group $D' = \{p_{16}, p_{17}, p_{18}, p_{19}, p_{20}\}$. Thus, $\succ_{\widehat{s'_1}}$ can



(a) 1-Type

(b) 2-Type, Crafted

(c) 2-Type, Random

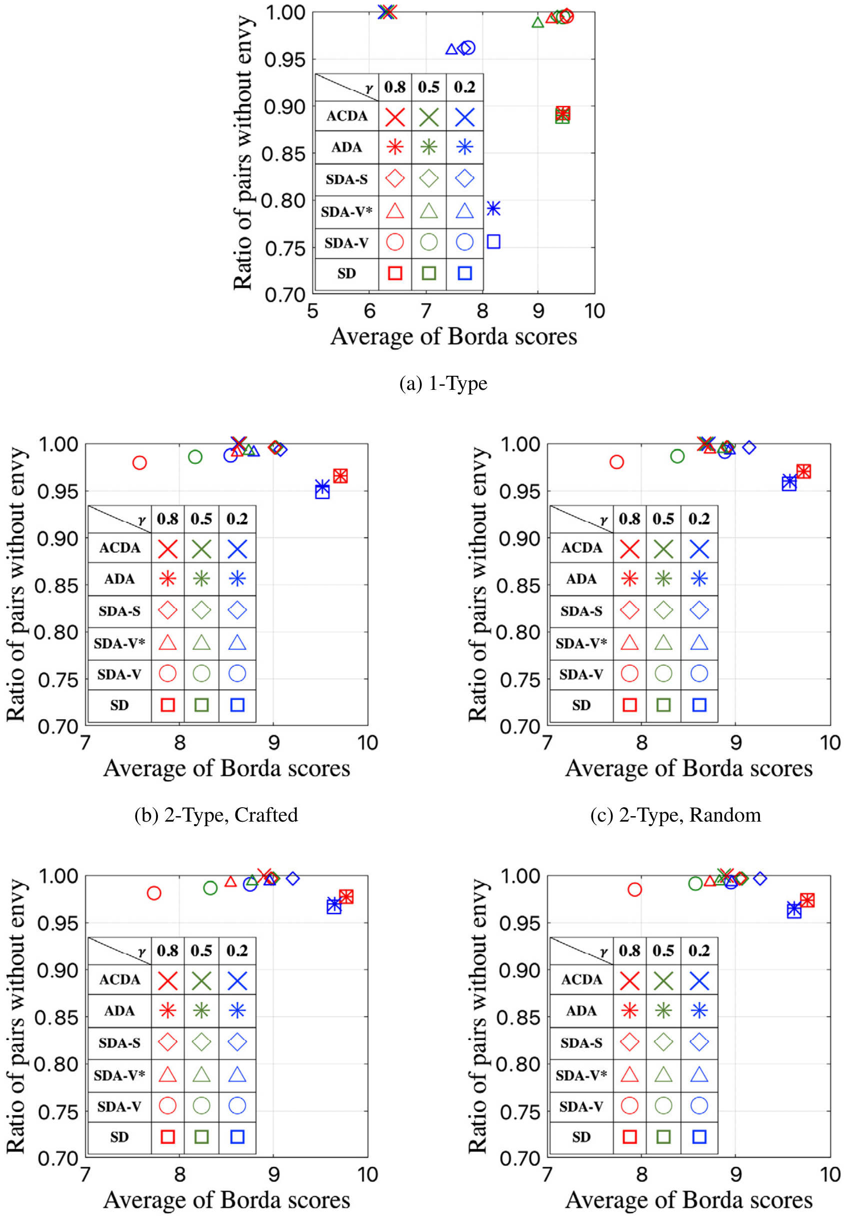
(d) 3-Type, Crafted

(e) 3-Type, Random

Fig. 3. Trade-off between efficiency and fairness ($\phi = 0.8$, $\rho = 0.1$, $\gamma = 0.2$, heterogeneous resources, Market I).

be represented as $A'B'C'D'$. Next, we create additional two central preferences, $\succ_{\widehat{s_5}'}$ and $\succ_{\widehat{s_6}'}$, which can be represented as $C'B'D'A'$ and $D'B'A'C'$, respectively. More precisely, we choose $\succ_{\widehat{s_5'}} = p_{11} \succ p_{12} \succ p_{13} \succ p_{14} \succ p_{15} \succ p_6 \succ p_7 \succ p_8 \succ p_9 \succ p_{10} \succ p_{16} \succ p_{17} \succ p_{18} \succ p_{19} \succ p_{20} \succ p_1 \succ p_2 \succ p_3 \succ p_4 \succ p_5$, and $\succ_{\widehat{s_6'}} = p_{16} \succ p_{17} \succ p_{18} \succ p_{19} \succ p_{20} \succ p_6 \succ p_7 \succ p_8 \succ p_9 \succ p_{10} \succ p_1 \succ p_2 \succ p_3 \succ p_4 \succ p_5 \succ p_{11} \succ p_{12} \succ p_{13} \succ p_{14} \succ p_{15}$. The population for each central preference is 133 or 134.

Random: We also examine the case where three central preferences are randomly chosen. For Market I, we use three randomly chosen central preferences: $\succ_{\widehat{s_7}} = p_{10} \succ p_3 \succ p_2 \succ p_5 \succ p_6 \succ p_1 \succ p_4 \succ p_9 \succ p_8 \succ p_7$, $\succ_{\widehat{s_8}} = p_7 \succ p_5 \succ p_{10} \succ p_1 \succ p_6 \succ p_2 \succ p_9 \succ p_4 \succ p_8 \succ p_3$, and $\succ_{\widehat{s_9}} = p_1 \succ p_9 \succ p_6 \succ p_4 \succ p_8 \succ p_2 \succ p_{10} \succ p_5 \succ p_7 \succ p_3$. For Market II, we use three randomly chosen central preferences: $\succ_{\widehat{s_7'}} = p_{20} \succ p_{17} \succ p_8 \succ p_{15} \succ p_{16} \succ p_{19} \succ p_{14} \succ p_4 \succ p_{11} \succ p_{13} \succ p_6 \succ p_7 \succ p_{10} \succ p_{12} \succ p_1 \succ p_5 \succ p_{18} \succ p_3 \succ p_2 \succ p_9$, $\succ_{\widehat{s_8'}} = p_1 \succ p_{20} \succ p_{14} \succ p_{13} \succ p_2 \succ p_{10} \succ p_5 \succ p_{12} \succ p_{17} \succ p_{16} \succ p_7 \succ p_3 \succ p_9 \succ p_8 \succ p_{18} \succ p_{15} \succ p_6 \succ p_{19} \succ p_4 \succ p_{11}$, and $\succ_{\widehat{s_9'}} = p_{16} \succ p_{19} \succ p_8 \succ p_3 \succ p_{15} \succ p_{14} \succ p_{17} \succ p_2 \succ p_{12} \succ p_{13} \succ p_1 \succ p_4 \succ p_5 \succ p_7 \succ p_{18} \succ p_6 \succ p_{20} \succ p_{10} \succ p_9 \succ p_{11}$.



(a) 1-Type

(b) 2-Type, Crafted

(c) 2-Type, Random

(d) 3-Type, Crafted

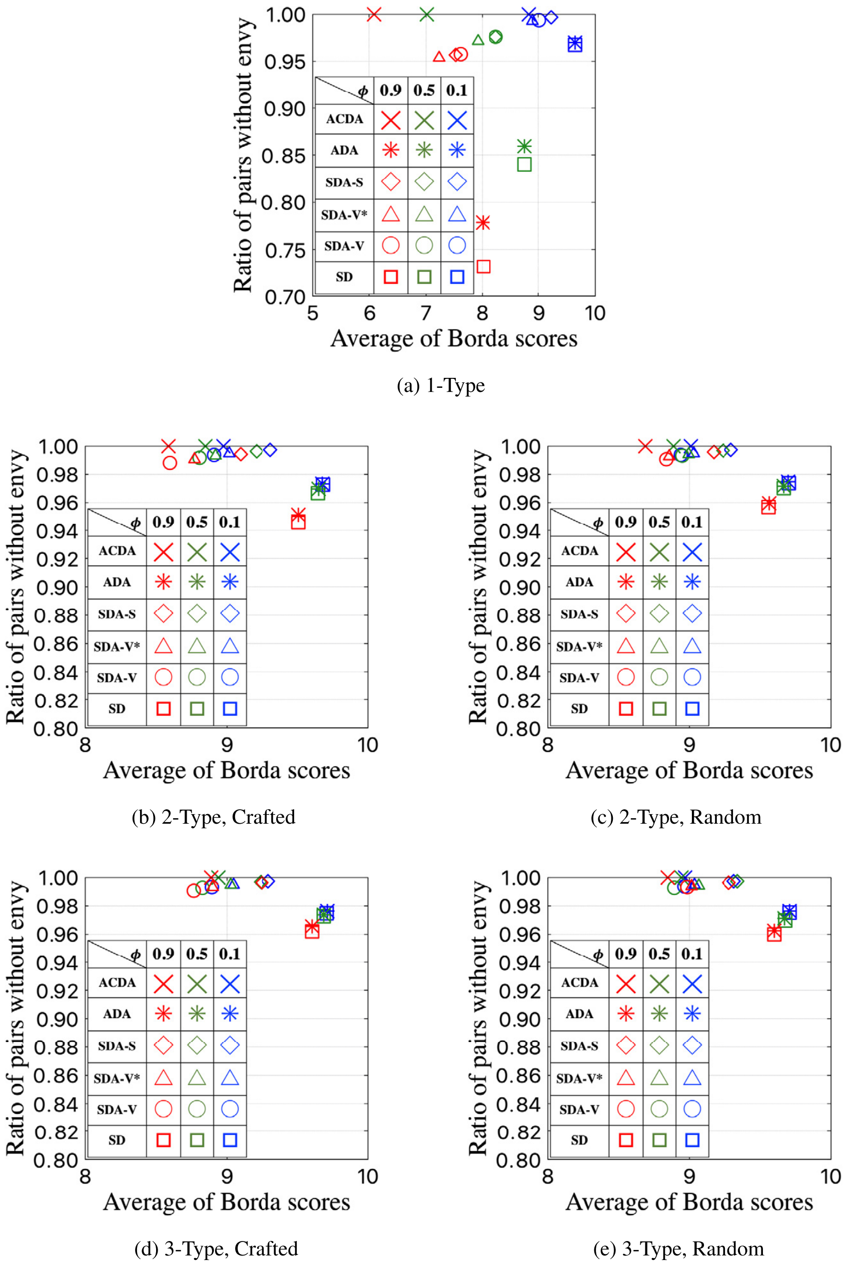
(e) 3-Type, Random

Fig. 4. Trade-off between efficiency and fairness by varying γ ($\phi = 0.8$, $\rho = 0.1$, $\gamma = 0.2, 0.5, 0.8$, homogeneous resources, Market I).

Table 2 summarizes the three types of preferences in two markets. The preference of each project $\succ_p$ is drawn uniformly at random. We create 100 instances for each parameter setting and compare three sample-based mechanisms (SDA-V, SDA-V*, SDA-S) with ACDA, SD, and ADA. ADA needs a capacity limit for each project p. As described earlier, we set this value to $\sum_{r|T_r \ni p} q_r$, which is the largest capacity when all of the shared resources are allocated to it. Since this capacity is large and not binding in many cases, ADA resembles SD. We use a Gurobi optimizer [59] to solve FEASIBILITY in these mechanisms.

5.2. Results

To illustrate the trade-off between efficiency and fairness, we plot the results of the obtained matching in a two-dimensional space in Figs. 2 to 7 where the x-axis shows the average Borda scores of the students; if a student is assigned to her i-th choice project, her score is $|P| - i + 1$, and the y-axis shows the ratio of the student pairs without any justified envy. Thus, the points located north-east are preferable. Each data point is an average for 100 problem instances for each mechanism.



(a) 1-Type

(b) 2-Type, Crafted

(c) 2-Type, Random

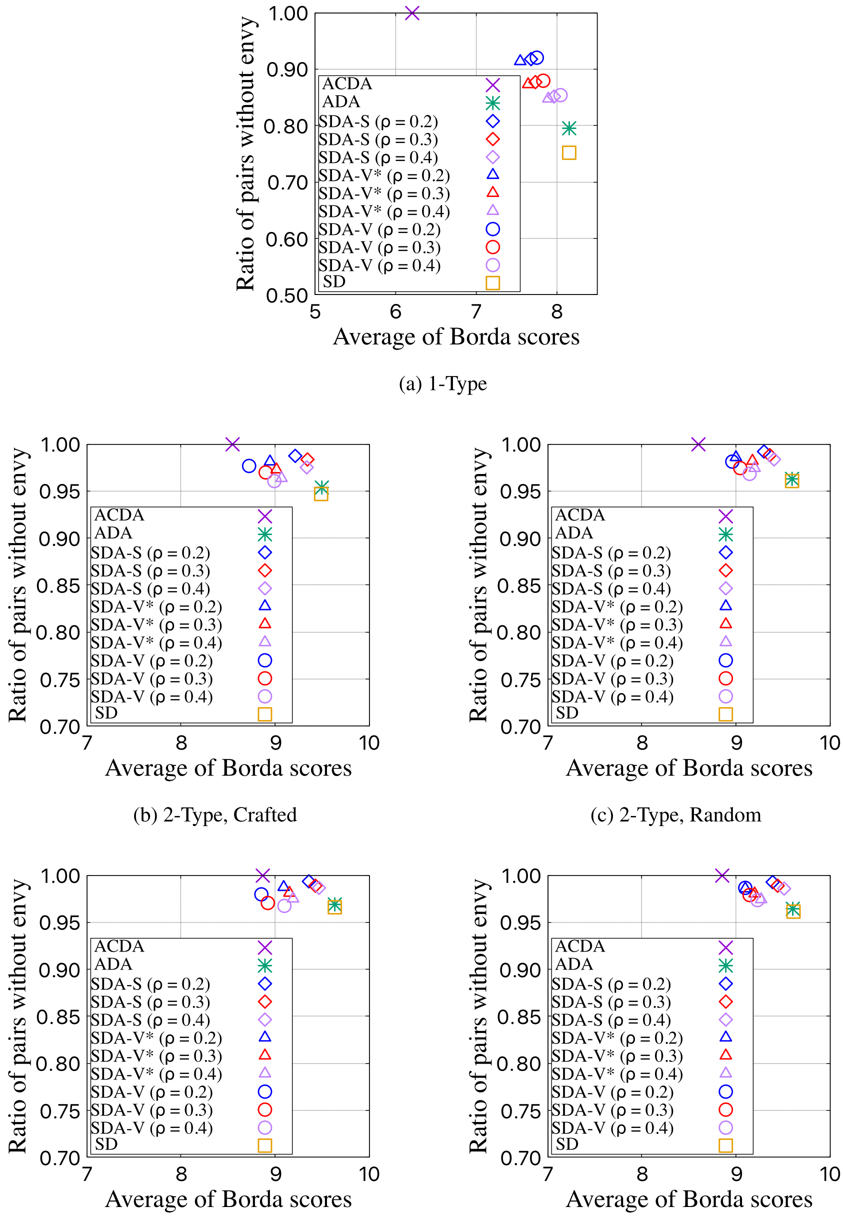
(d) 3-Type, Crafted

(e) 3-Type, Random

Fig. 5. Trade-off between efficiency and fairness by varying ϕ ($\rho = 0.1$, $\gamma = 0.2$, $\phi = 0.1, 0.5, 0.9$, homogeneous resources, Market I).

Table 2
Summary of three types of preferences in two markets.

Markets \ Cases				1-Type	2-Type		3-Type	
	$\lvert S\rvert$	$\lvert P\rvert$	$\lvert R\rvert$		Crafted	Random	Crafted	Random
I	200	10	20	$\succ_{\hat{s}_1}$	$\succ_{\hat{s}_1}, \succ_{\hat{s}_2}$	$\succ_{\hat{s}_3}, \succ_{\hat{s}_4}$	$\succ_{\hat{s}_1}, \succ_{\hat{s}_5}, \succ_{\hat{s}_6}$	$\succ_{\hat{s}_7}, \succ_{\hat{s}_8}, \succ_{\hat{s}_9}$
II	400	20	40	$\succ_{\hat{s}'_1}$	$\succ_{\hat{s}'_1}, \succ_{\hat{s}'_2}$	$\succ_{\hat{s}'_3}, \succ_{\hat{s}'_4}$	$\succ_{\hat{s}'_1}, \succ_{\hat{s}'_5}, \succ_{\hat{s}'_6}$	$\succ_{\hat{s}'_7}, \succ_{\hat{s}'_8}, \succ_{\hat{s}'_9}$



(a) 1-Type

(b) 2-Type, Crafted

(c) 2-Type, Random

(d) 3-Type, Crafted

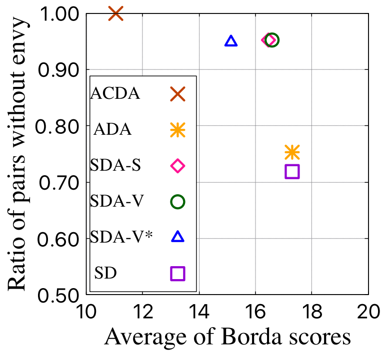
(e) 3-Type, Random

Fig. 6. Trade-off between efficiency and fairness by varying ρ ($\phi = 0.8$, $\gamma = 0.2$, $\rho = 0.2, 0.5, 0.8$, homogeneous resources, Market I).

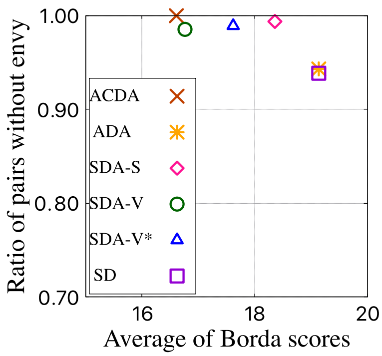
Figs. 2 (a)–(e) show the average performance for the 1-, 2-, and 3-types, where ϕ (the spread parameter in the Mallows model) is 0.8, γ (the probability that project p is included in T_r) is 0.2, and ρ (the ratio of sampled students) is 0.1. Also, resources are homogeneous, i.e., the capacity of each resource is 10.

Fig. 2 (a) shows the result for the 1-type case. The performances of the SDA mechanisms are similar and strike a better balance between fairness and efficiency than the existing mechanisms. In particular, the improvement of efficiency (students' welfare) is significant compared to ACDA. Note that to increase the average Borda score by one point, every student must be assigned to a strictly better project.

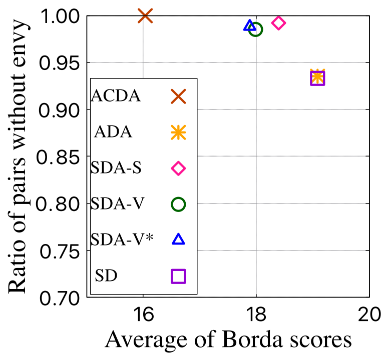
Figs. 2 (b) and (c) show the results for the 2-type crafted case and 2-type random case, respectively, and Figs. 2 (d) and (e) show the results for the 3-type crafted case and 3-type random case, respectively. For the crafted central preferences (Figs. 2 (b) and (d)), SDA-V* outperforms SDA-V, whose performance is slightly worse than ACDA. This is because these central preferences are chosen such that group B projects are likely to be winners, even though they are sub-optimal for all students. On the other hand, for random central preferences (Figs. 2 (c) and (e)), the performances of SDA-V* and



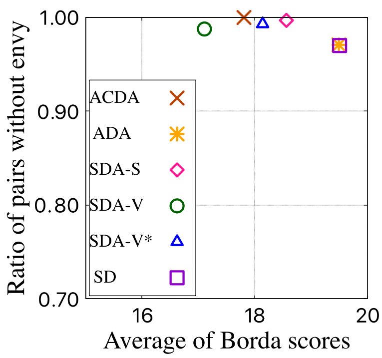
(a) 1-Type



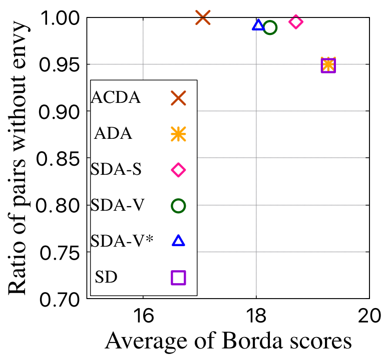
(b) 2-Type, Crafted



(c) 2-Type, Random



(d) 3-Type, Crafted



(e) 3-Type, Random

Fig. 7. Trade-off between efficiency and fairness ($\phi = 0.8$, $\rho = 0.1$, $\gamma = 0.2$, homogeneous resources, Market II).

SDA-V are very close and surpass ACDA. SDA-S outperforms SDA-V and SDA-V* both in the crafted/random central preferences.

For the 2/3-type cases, the students' preferences are more diverse. As a result, the competition among students becomes less severe; both the Borda score and the ratio of pairs without envy increase compared to the 1-type case.

Figs. 3 (a)–(e) show the average performance in a similar setting as Fig. 2, except that the resources are heterogeneous, i.e., the capacity of a resource is either $1, 5, 10, 15$, or 20 (there are four resources for each capacity). The obtained results are quite similar to Fig. 2. Thus, in the following experiments, we use homogeneous resources.

Figs. 4 (a)–(e) show the average performance by varying γ as $0.2, 0.5$, and 0.8; the other parameters are set as follow: $\phi = 0.8$, $\rho = 0.1$, and homogeneous resources with $q_r = 10$ for all $r \in R$. When γ becomes larger, we have more freedom to allocate resources. As a result, students' welfare increases as long as a mechanism can allocate resources wisely according to students' demands. Fig. 4 (a) clearly shows that students' welfare improves as γ increases for all mechanisms except ACDA. Furthermore, the performance of SDA-V degrades in Fig. 4 for 2/3-type cases. This result indicates choosing a single winner by the Borda count is inappropriate when students are divided into multiple types.

Figs. 5 (a)–(e) show the average performance by varying ϕ as 0.1, 0.5, and 0.9; the other parameters are set as follow: $\rho = 0.1$, $\gamma = 0.2$, and homogeneous resources with $q_r = 10$ for all $r \in R$. When ϕ is large, the students' preferences are almost identical. Thus, the competition among students becomes more severe and resource allocation significantly affects their welfare. When ϕ is small, the difference among mechanisms becomes smaller. SDA-V, SDA-V* and SDA-S are quite close when the students' preferences are identical, while SDA-S outperforms SDA-V in diverse cases. In particular, the performance of SDA-V can be worse than ACDA.

One might argue that SDA mechanisms work only when the sampled students resemble regular students. Although this is true to some extent, our result shows that when student preferences are diverse, all the mechanisms work reasonably well.

Figs. 6 (a)–(e) show the average performance by varying ρ as 0.2, 0.3, and 0.4 for SDA mechanism; the other parameters are set as follow: $\phi = 0.8$, $\gamma = 0.2$, and homogeneous resources with $q_r = 10$ for all $r \in R$. When ρ is small, these mechanisms resemble ACDA. By increasing ρ, they gradually become similar to SD. Thus, by controlling parameter ρ, we can further fine-tune the balance. SDA-S outperforms SDA-V and SDA-V* for all ρ.

Figs. 7 (a)–(e) show the results for Market II ($\phi = 0.8$, $\rho = 0.1$, $\gamma = 0.2$, homogeneous resources $q_r = 10$ for all $r \in R$). The qualitative trends are basically very similar to Market I, while the differences in the Borda scores among mechanisms become larger.

6. Conclusion

We introduced a student-to-project matching problem that endogenously handles the resource allocation problem that defines the capacity of projects. We showed that it is impossible to design a mechanism that is fair, strategyproof, and satisfies very mild efficiency properties. To strike a good balance between fairness and efficiency, we developed a new class of strategyproof mechanisms called sampled-based Deferred Acceptance (SDA), and proved that it is weakly nonwasteful, fair among some students, weakly Pareto efficient, and weakly group strategyproof. We also developed three SDA instances called SDA-V, SDA-V*, and SDA-S, and proved that they are resource efficient. We also showed that they satisfy Pareto efficiency and fairness under some special cases and experimentally evaluated these mechanisms. Our experimental results show that SDA-S outperforms the other two mechanisms in terms of efficiency (students' welfare) and fairness when students are divided into different types according to their preferences.

Our future works will theoretically identify the optimal sample size and deal with a case where various constraints are imposed on the allocation of resources, e.g., where the total number of resources that can be allocated to each project is bounded. We will also examine a combined model of an SPR and other generalizations of two-sided matching, such as a matching with budget constraints [12,66] and a matching with multidimensional resources (a.k.a. a refugee resettlement problem) [67,68].

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Kwei-guu Liu reports financial support was provided by Toyota Motor Corporation.

Data availability

No data was used for the research described in the article.

Acknowledgements

This work is partially supported by the joint project with Toyota Motor Corporation, titled "Advanced Mathematical Science for Mobility Society", and JSPS KAKENHI Grant Numbers JP20H00609 and JP21H04979.

References

[1] K. Yahiro, M. Yokoo, Game theoretic analysis for two-sided matching with resource allocation, in: Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2020), 2020, pp. 1548–1556.
[2] M. Goto, F. Kojima, R. Kurata, A. Tamura, M. Yokoo, Designing matching mechanisms under general distributional constraints, Am. Econ. J. Microecon. 9 (2) (2017) 226–262.
[3] A.E. Roth, M.A.O. Sotomayor, Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis, Econometric Society Monographs, Cambridge University Press, 1990.
[4] B. Korte, J. Vygen, Combinatorial Optimization: Theory and Algorithms, Springer, 2018.
[5] D. Gale, L.S. Shapley, College admissions and the stability of marriage, Am. Math. Mon. 69 (1) (1962) 9–15.
[6] Y. Okumura, School choice with general constraints: a market design approach for the nursery school waiting list problem in Japan, Jpn. Econ. Rev. (2019) 497–516.
[7] H. Aziz, P. Biró, T. Fleiner, S. Gaspers, R. de Haan, N. Mattei, B. Rastegari, Stable matching with uncertain pairwise preferences, Theor. Comput. Sci. 909 (2022) 1–11.
[8] H. Aziz, S. Gaspers, Z. Sun, T. Walsh, From matching with diversity constraints to matching with regional quotas, in: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2019), 2019, pp. 377–385.

[9] H. Hosseini, K. Larson, R. Cohen, On manipulablity of random serial dictatorship in sequential matching with dynamic preferences, in: Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI 2015), 2015, pp. 4168–4169.
[10] Y. Kawase, A. Iwasaki, Near-feasible stable matchings with budget constraints, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI 2017), 2017, pp. 242–248.
[11] K. Yahiro, Y. Zhang, N. Barrot, M. Yokoo, Strategyproof and fair matching mechanism for ratio constraints, Auton. Agents Multi-Agent Syst. 34 (2020) 1–29.
[12] A. Ismaili, N. Hamada, Y. Zhang, T. Suzuki, M. Yokoo, Weighted matching markets with budget constraints, J. Artif. Intell. Res. 65 (2019) 393–421.
[13] T. Suzuki, A. Tamura, K. Yahiro, M. Yokoo, Y. Zhang, Strategyproof allocation mechanisms with endowments and M-convex distributional constraints, Artif. Intell. 315 (2023) 103825.
[14] F. Kojima, A. Tamura, M. Yokoo, Designing matching mechanisms under constraints: an approach from discrete convex analysis, J. Econ. Theory 176 (2018) 803–833.
[15] Y. Kamada, F. Kojima, Fair matching under constraints: theory and applications, Rev. Econ. Stud. (2020), in press.
[16] Y. Kamada, F. Kojima, Efficient matching under distributional constraints: theory and applications, Am. Econ. Rev. 105 (1) (2015) 67–99.
[17] D. Fragiadakis, A. Iwasaki, P. Troyan, S. Ueda, M. Yokoo, Strategyproof matching with minimum quotas, ACM Trans. Econ. Comput. 4 (1) (2016) 6.
[18] M. Goto, A. Iwasaki, Y. Kawasaki, R. Kurata, Y. Yasuda, M. Yokoo, Strategyproof matching with regional minimum and maximum quotas, Artif. Intell. 235 (2016) 40–57.
[19] N. Hamada, C. Hsu, R. Kurata, T. Suzuki, S. Ueda, M. Yokoo, Strategy-proof school choice mechanisms with minimum quotas and initial endowments, Artif. Intell. 249 (2017) 47–71.
[20] T. Sönmez, T.B. Switzer, Matching with (branch-of-choice) contracts at the United States Military Academy, Econometrica 81 (2) (2013) 451–488.
[21] T. Sönmez, Bidding for army career specialties: improving the ROTC branching mechanism, J. Polit. Econ. 121 (1) (2013) 186–219.
[22] I.E. Hafalir, M.B. Yenmez, M.A. Yildirim, Effective affirmative action in school choice, Theor. Econ. 8 (2) (2013) 325–363.
[23] L. Ehlers, I.E. Hafalir, M.B. Yenmez, M.A. Yildirim, School choice with controlled choice constraints: hard bounds versus soft bounds, J. Econ. Theory 153 (2014) 648–683.
[24] F. Kojima, School choice: impossibilities for affirmative action, Games Econ. Behav. 75 (2) (2012) 685–693.
[25] R. Kurata, N. Hamada, A. Iwasaki, M. Yokoo, Controlled school choice with soft bounds and overlapping types, J. Artif. Intell. Res. 58 (2017) 153–184.
[26] K. Tomoeda, Finding a stable matching under type-specific minimum quotas, J. Econ. Theory 176 (2018) 81–117.
[27] P. Biró, T. Fleiner, R.W. Irving, D.F. Manlove, The college admissions problem with lower and common quotas, Theor. Comput. Sci. 411 (34–36) (2010) 3136–3153.
[28] T. Fleiner, N. Kamiyama, A matroid approach to stable matchings with lower quotas, Math. Oper. Res. 41 (2) (2016) 734–744.
[29] K. Hamada, K. Iwama, S. Miyazaki, The hospitals/residents problem with lower quotas, Algorithmica 74 (1) (2016) 440–465.
[30] D. Gale, L.S. Shapley, College admissions and the stability of marriage, Am. Math. Mon. 69 (1) (1962) 9–15.
[31] R.W. Irving, D.F. Manlove, G. O'Malley, Stable marriage with ties and bounded length preference lists, J. Discret. Algorithms 7 (2) (2009) 213–219.
[32] D.F. Manlove, The structure of stable marriage with indifference, Discrete Appl. Math. 122 (1–3) (2002) 167–181.
[33] O. Aygün, B. Turhan, Dynamic reserves in matching markets, J. Econ. Theory 188 (2020) 105069.
[34] H. Aziz, A. Baychkov, P. Biró, Cutoff stability under distributional constraints with an application to summer internship matching, Math. Program. (2022) 1–23.
[35] H. Aziz, B. Klaus, Random matching under priorities: stability and no envy concepts, Soc. Choice Welf. 53 (2) (2019) 213–259.
[36] P. Krishnaa, G. Limaye, M. Nasre, P. Nimbhorkar, Envy-freeness and relaxed stability: hardness and approximation algorithms, J. Comb. Optim. 45 (2023) 41.
[37] K.C. Ágoston, P. Biró, E. Kováts, Z. Jankó, College admissions with ties and common quotas: integer programming approach, Eur. J. Oper. Res. 299 (2) (2022) 722–734.
[38] A. Ismaili, T. Yamaguchi, M. Yokoo, Student-project-resource allocation: complexity of the symmetric case, in: Proceedings of the 21st International Conference on Principles and Practice of Multi-Agent Systems (PRIMA 2018), 2018, pp. 226–241.
[39] A. Alkan, Nonexistence of stable threesome matchings, Math. Soc. Sci. 16 (2) (1988) 207–209.
[40] C.-C. Huang, Two's company, three's a crowd: stable family and threesome roommates problems, in: Proceedings of the 15th Annual European Symposium on Algorithms (ESA 2007), 2007, pp. 558–569.
[41] C. Ng, D.S. Hirschberg, Three-dimensional stable matching problems, SIAM J. Discrete Math. 4 (2) (1991) 245–252.
[42] R. Bredereck, K. Heeger, D. Knop, R. Niedermeier, Multidimensional stable roommates with master list, in: Proceedings of the 16th International Conference on Web and Internet Economics (WINE 2020), Springer, 2020, pp. 59–73.
[43] M. McKay, D. Manlove, The three-dimensional stable roommates problem with additively separable preferences, in: Proceedings of the 14th International Symposium on Algorithmic Game Theory (SAGT 2021), Springer, 2021, pp. 266–280.
[44] D.J. Abraham, R.W. Irving, D.F. Manlove, Two algorithms for the student-project allocation problem, J. Discret. Algorithms 5 (1) (2007) 73–90.
[45] P. Biró, E. McDermid, Matching with sizes (or scheduling with processing set restrictions), Discrete Appl. Math. 164 (2014) 61–67.
[46] C.A. Glass, H. Kellerer, Parallel machine scheduling with job assignment restrictions, Nav. Res. Logist. 54 (3) (2007) 250–257.
[47] J.W. Hatfield, P.R. Milgrom, Matching with contracts, Am. Econ. Rev. 95 (4) (2005) 913–935.
[48] Y. Kamada, F. Kojima, Stability concepts in matching under distributional constraints, J. Econ. Theory 168 (2017) 107–142.
[49] S.-H. Cho, M. Koshimura, P. Mandal, K. Yahiro, M. Yokoo, Impossibility of weakly stable and strategy-proof mechanism, Econ. Lett. 217 (2022) 110675.
[50] R.M. Karp, Reducibility among combinatorial problems, in: R.E. Miller, J.W. Thatcher, J.D. Bohlinger (Eds.), Complexity of Computer Computations, Springer, 1972, pp. 85–103.
[51] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, third edition, The MIT Press, 2009.
[52] A.V. Goldberg, J.D. Hartline, A. Wright, Competitive auctions and digital goods, in: Proceedings of the 12th Annual Symposium on Discrete Algorithms (SODA 2001), 2001, pp. 735–744.
[53] C. Borgs, J. Chayes, N. Immorlica, M. Mahdian, A. Saberi, Multi-unit auctions with budget-constrained bidders, in: Proceedings of the 6th ACM Conference on Electronic Commerce (EC 2005), 2005, pp. 44–51.
[54] L.E. Dubins, D.A. Freedman, Machiavelli and the Gale-Shapley algorithm, Am. Math. Mon. 88 (7) (1981) 485–494.
[55] A.E. Roth, The economics of matching: stability and incentives, Math. Oper. Res. 7 (4) (1982) 617–628.
[56] L.-G. Svensson, Strategy-proof allocation of indivisible goods, Soc. Choice Welf. 16 (4) (1999) 557–567.
[57] S. Pápai, Strategyproof assignment by hierarchical exchange, Econometrica 68 (6) (2000) 1403–1433.
[58] S. Barberà, D. Berga, B. Moreno, Group strategy-proofness in private good economies, Am. Econ. Rev. 106 (4) (2016) 1073–1099.
[59] Gurobi, Gurobi optimization, http://www.gurobi.com/, 2022.
[60] C.L. Mallows, Non-null ranking models. I, Biometrika 44 (1–2) (1957) 114–130.
[61] J.D. Tubbs, Distance based binary matching, in: Computing Science and Statistics, Springer, 1992, pp. 548–550.
[62] T. Lu, C. Boutilier, Effective sampling and learning for mallows models with pairwise-preference data, J. Mach. Learn. Res. 15 (2014) 3963–4009.
[63] J. Drummond, C. Boutilier, Elicitation and approximately stable matching with partial preferences, in: Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013), 2013, pp. 97–105.

[64] W.V. Gehrlein, P.C. Fishburn, The probability of the paradox of voting: a computable solution, J. Econ. Theory 13 (1) (1976) 14–25.
[65] N. Mattei, T. Walsh, Preflib: a library for preferences, in: Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT 2013), 2013, pp. 259–270, http://www.preflib.org.
[66] A. Abizada, Stability and incentives for college admissions with budget constraints, Theor. Econ. 11 (2) (2016) 735–756.
[67] H. Aziz, J. Chen, S. Gaspers, Z. Sun, Stability and Pareto optimality in refugee allocation matchings, in: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2018), 2018, pp. 964–972.
[68] D. Delacretaz, S.D. Kominers, A. Teytelboym, Matching mechanisms for refugee resettlement, working paper, 2020.