Game Theoretic Analysis for Two-Sided Matching with Resource Allocation*

Kentaro Yahiro
Kyushu University
Fukuoka, Japan
yahiro@agent.inf.kyushu-u.ac.jp

Makoto Yokoo†
Kyushu University
Fukuoka, Japan
yokoo@inf.kyushu-u.ac.jp

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. Although students are many-to-one matched to projects, indivisible resources are many-to-one allocated 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 (resource allocation problem), and (2) students are matched to projects based on the capacities determined in the previous problem (matching problem). Although both problems are well-understood, if the expectations used in the first are incorrect, we obtain a suboptimal outcome. Thus, this problem must be solved as a whole without dividing it in two parts. We show that no strategyproof mechanism satisfies fairness (i.e., no student has justified envy) and weak efficiency requirements on students' welfare. Given this impossibility result, we develop a new strategyproof mechanism that strikes a good balance between fairness and efficiency and assess it by experiments.

KEYWORDS

Matching, Resource Allocation, Strategyproofness

ACM Reference Format:
Kentaro Yahiro and Makoto Yokoo. 2020. Game Theoretic Analysis for Two-Sided Matching with Resource Allocation. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 9 pages.

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 [44] since students are matched to projects based on their preferences. On the other hand, it is also a discrete resource allocation problem [33] since resources are allocated to each project. However, unlike the standard two-sided matching setting, where the capacity of each project is exogenously determined, we assume the capacities are endogenously determined by resource allocation.

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 their preferences beforehand, she can find a matching that satisfies desirable properties (e.g., stability) with a strategyproof mechanism, e.g., Deferred Acceptance mechanism (DA) [13], such that students voluntarily disclose their true preferences. However, the mechanism designer usually does not know their preferences. Thus, a common practice is to determine the resource allocation part based on the expectations or the past data and to 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 suboptimal; excess demand and supply for seats may coexist in the same matching-allocation, which can be resolved by a better resource allocation.

One real-life instance where this practice is applied is the nursery-school waiting list problem [42]. 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 boost the number of nursery schools to encourage more 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 in 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, which is determined in advance. Assume a local government (e.g., a city/prefecture/state) has spare resources, e.g.,

*An extended abstract of an earlier version of this work was presented [26].
†Also with RIKEN, Center for Advanced Intelligence Project AIP.
[1] An SPR is a strict generalization of the student-project-room allocation problem [17], 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.

[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. An SPR is more general and can represent a variety of applications beyond this example.

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

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 the spare budget/resources.

Our main contribution is presenting a generalized framework to capture two orthogonal problems that need to be solved simultaneously as a single problem by introducing SPR. Then we show several impossibility results, in particular, that no strategyproof mechanism satisfies fairness (i.e., no student has justified envy) and weak efficiency requirements on students' welfare. We also confirm the limitations of the following three existing mechanisms [17]: Serial Dictatorship mechanism (SD), Artificial Caps Deferred Acceptance mechanism (ACDA), and Adaptive Deferred Acceptance mechanism (ADA). Then we introduce a new strategyproof mechanism called Sample and Vote Deferred Acceptance (SVDA), which satisfies several properties on fairness and efficiency. SVDA 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.

The rest of this paper is organized as follows. Section 2 introduces a model of an SPR. Section 3 proves several impossibility results. In Section 4, we discuss the existence of strategyproof mechanisms and introduce our new strategyproof mechanism, SVDA. Next, we numerically show that it strikes a good balance between fairness and efficiency in Section 5. Finally, Section 6 concludes our paper.

1.1 Related Work

This paper follows previous works that address constrained matching problems. Two-sided matching has attracted considerable attention from AI and theoretical computer science researchers [3, 4, 23, 25, 30, 49, 50]. 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 constraints on the legitimate distribution of students over schools (i.e., distributional constraints) [32], including regional maximum quotas, which restrict the total number of students assigned to a set of schools [28], minimum quotas, which guarantee that a certain number of students are assigned to each school [12, 16, 21, 45, 46], and diversity constraints [10, 19, 31, 34, 47]. Other works examine the computational complexity for finding a matching with desirable properties under distributional constraints [6, 11, 20]. Ismaili et al. [27] deal with a similar model, but their model utilizes a compact representation scheme to handle exponentially many students assuming they can be divided into a small number of types.

Several works exist on three-sided matching problems [2, 24, 41] where three types of players/agents are matched, e.g., males, females, and pets. Although their model superficially resembles 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 (SPA) [1], students are matched to projects, which are offered by lecturers. A student has a preference over projects, and a lecturer has a preference over students. Each lecturer has a capacity limit. An SPR can be formalized as a two-sided matching problem with distributional constraints [32]. There also exists an alternative model of SPA, where a lecturer has a preference over projects [38, 39].

2 MODEL

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

Definition 2.1 (Student-Project-Resource allocation (SPR) Instance). An SPR instance is a tuple $(S, P, R, \succ_S, \succ_P, T_R, q_R)$.

- $S = \{s_1, \dots, s_{|S|}\}$ is a set of students.
- $P = \{p_1, \dots, p_{|P|}\}$ is a set of projects.
- $R = \{r_1, \dots, r_{|R|}\}$ 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 project 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 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.

We illustrate our setting with the following example.

Example 2.2. 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 the preferences:

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

Since we assume resources are indivisible, it is impossible to allocate students to three different projects although the total capacity of all resources equals three. A resource can be allocated only 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 $(0, 2, 0, 1)$.

We follow the matching with contracts model [22]. 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 which satisfy the following conditions.

Definition 2.3 (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.

[3]In designing a strategyproof mechanism, we assume student preferences are private information, and the other information is public. Thus, X can be characterized by public information.

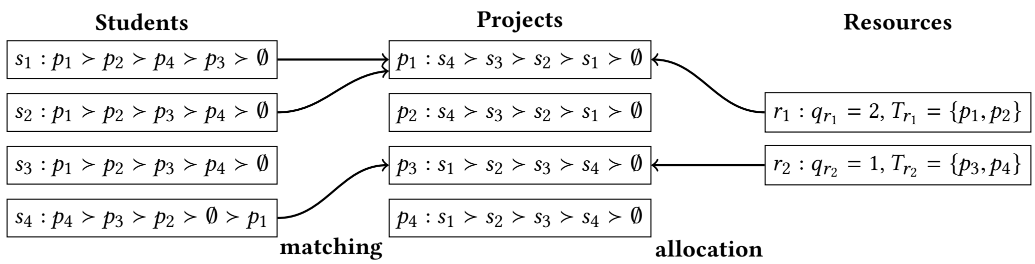


Figure 1: SPR instance: matching $\hat{Y}$ and allocation $\hat{\mu}$ in Example 2.2

For matching Y, let $Y(s) \in P \cup \{\emptyset\}$ 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. In an SPR, we also need to describe how resources are allocated to projects. The *feasiblity* of a matching is defined based on this description.

Definition 2.4 (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]

Definition 2.5 (Feasibility). A feasible matching (Y, μ) is a matching-allocation pair where $|Y(p)| \leq q_\mu(p)$ holds for every $p \in P$.

In Example 2.2, assume matching $\hat{Y}$ is $\{(s_1, p_1), (s_2, p_1), (s_3, \emptyset), (s_4, p_3)\}$ and allocation $\hat{\mu}$ is distributing r_1 to p_1, and r_2 to p_3. Then $(\hat{Y}, \hat{\mu})$ is a feasible matching. See Figure 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 2.6 (Claiming an Empty Seat with μ). For 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 2.7 (Nonwastefulness). For 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'. Feasible matching (Y, μ) is nonwasteful if no student possibly claims an empty seat.

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

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

In the setting of Example 2.2 (Figure 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 2.8 (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 2.2 in Figure 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 2.9 (Stability). A feasible matching (Y, μ) is stable if it is nonwasteful and fair.

Next we introduce concepts on students' welfare (efficiency).

Definition 2.10 (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 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 2.2, matching $\hat{Y}$ in Figure 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.

Definition 2.11 (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 2.12 ((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 3.1. *An SPR instance exists where no feasible matching is fair and nonwasteful.*

Proof. Consider the following SPR instance: two students, s_1, s_2, two projects, p_1, p_2, and a unitary resource compatible with both. The student preferences are $p_1 \succ_{s_1} p_2$ and $p_2 \succ_{s_2} p_1$. The project preferences are $s_2 \succ_{p_1} s_1$ and $s_1 \succ_{p_2} s_2$. By symmetry, we can assume the resource is allocated to p_1 w.l.o.g. From 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 3.2 (Weak Nonwastefulness). Feasible matching (Y, μ) is weakly nonwasteful if no student claims an empty seat with current allocation μ.

In the setting of Example 2.2, feasible matching $(\hat{Y}, \hat{\mu})$ in Figure 1 is weakly nonwasteful because no student can be assigned to a better project with current allocation $\hat{\mu}$.

Definition 3.3 (Very Weak Nonwastefulness). For feasible matching (Y, μ), student s *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 strongly claims an empty seat.

In other words, student s 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 2.2. Then s_3 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 strongly claims an empty seat, she is currently unassigned and claims an empty seat in project p with current allocation μ. If she claims an empty seat in p with 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 3.4 (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 3.5 (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 3.6. *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 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 is allocated to p' while $p, p' \in T_r$ holds. 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 the fact that Y is Pareto efficient. □

Now we are ready to introduce another impossibility theorem.

Theorem 3.7. *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 the 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 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 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* [17]. 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. Goto et al. [17] present three general strategyproof mechanisms in this context. Since an SPR satisfies heredity, the properties of these mechanisms are automatically inherited to our model.

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

Definition 4.1 (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 the reduction from a partition problem, which is known to be NP-complete [29].

Definition 4.2 (Partition). Can a given multiset $V = \{v_1, \ldots, v_\ell\}$ of positive integers be partitioned into two multisets V_1 and V_2 such that the sum of the numbers in V_1 equals the sum of the numbers in V_2?

Theorem 4.3. *Feasibility is NP-complete.*

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 Partition can be reduced to an SPR instance. For integer multiset $V = \{v_1, \ldots, v_\ell\}$, such that $\sum_{v \in V} v = 2m$, we create two projects, p_1 and p_2, and ℓ resources $R = \{r_1, \ldots, r_\ell\}$. Each resource $r_i \in R$ is compatible with both projects, i.e., $T_{r_i} = \{p_1, p_2\}$, and the capacity q_{r_i} equals to v_i. We assume $2m$ students, and in given matching Y, m students are matched to both projects p_1 and p_2. Clearly, if the original Partition is *yes*, i.e., if we can partition V into two multisets that hit m, then the corresponding Feasiblity is *yes*, and vice versa.

Since Partition is NP-complete, Feasibility is also NP-complete. □

To verify feasibility, we need to solve a Mixed Integer Programming (MIP) instance. For a special case where $q_r = 1$ for all r and T_R has a laminar structure, the distributional constraints form an $M^\natural$-convex set; a generalized mechanism can obtain a fair matching [32] since Feasibility is no longer NP-complete.

In the following, we describe these mechanisms adopted to an SPR one by one. First, Serial Dictatorship mechanism (SD) uses a serial order among students. The order can be arbitrary, but it must be determined independently from 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 2.2, s_1 will choose $\boldsymbol{Y}_1 = \{Y \in \boldsymbol{Y} \mid (s_1, p_1) \in Y\}$, i.e., all elements in $\boldsymbol{Y}$ such that s_1 is allocated to p_1. Then 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 2.2, s_2 will choose $\boldsymbol{Y}_2 = \{Y \in \boldsymbol{Y}_1 \mid (s_2, p_1) \in Y\}$, i.e., all 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 exogenously determined possibilities. SD is also Pareto efficient by the following reason. Clearly, we cannot improve the assignment of s_1. Moreover, we cannot improve s_2 without hurting s_2, 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 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 computationally expensive in our setting since we need to solve Feasibility for $Y \cup \{(s_k, p)\}$ in Step 2. Nor does SD satisfy fairness. 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) [13]. 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 among 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 from student preferences and decide maximum quotas based on it.

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

Mechanism 2 (Artificial Caps Deferred Acceptance (ACDA)).

Step 1: *Choose μ independently from $\succ_S$.*

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 from their preferences.

The third mechanism is Adaptive Deferred Acceptance (ADA). Like SD, ADA uses serial order among students. The order can be arbitrary, but it must be determined independently from 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 the identical matching as SD. To apply ADA to an SPR, we choose q_p as $\sum_{r \mid T_r \ni p} q_r$, which is the largest capacity when all compatible resources are allocated to it. During the execution of ADA, project p is forbidden under (partial) matching Y if no allocation μ exists with which $Y \cup \{(s,p)\}$ becomes feasible even though $|Y(p)|$ is strictly less than the (current) maximum quota of p. In other words, project p is forbidden if p cannot accept another student due to resource contention among projects. Formally, ADA for an SPR is defined as follows:

Mechanism 3 (Adaptive Deferred Acceptance (ADA)). *We initially assume 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 2.2 The maximum quotas of 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, we cannot allocate another student to it since r_1 is 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 already reaches 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 by using DA. Then, project p_4 is forbidden. Although its maximum quota is one and no student is currently assigned, 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 student 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.

Table 1: Mechanism Properties

	Fairness	PE	NW	Weak NW	RE
SD [17]	no	yes	yes	yes	yes
ADA [17]	no	no	yes	yes	yes
SVDA	no	no	no	yes	yes
ACDA [17]	yes	no	no	yes	no

Using these mechanisms, we can show that resource efficiency and very weak nonwastefulness are independent properties. For example, ACDA satisfies very weak nonwastefulness. Under resource allocation μ in ACDA, no student claims an empty seat with μ, although it does not satisfy resource efficiency, i.e., it may allocate a resource to a unanimously less preferred project. Next assume a mechanism does not assign any 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 (including the new mechanism introduced in Section 4.2), where PE stands for Pareto efficiency, NW for nonwastefulness, and RE for resource efficiency.

4.2 Sample and Vote Deferred Acceptance Mechanism

ACDA is too inefficient, and SD and ADA are too unfair (many students have justified envy) and computationally expensive (feasibility must be verified $O(|S \times P|)$ times). Moreover, Theorem 3.7 shows that fairness cannot be achieved without significantly sacrificing efficiency. In this section, we introduce a new strategyproof mechanism called Sample and Vote Deferred Acceptance (SVDA), which strikes a good balance between fairness and efficiency by slightly sacrificing fairness to improve efficiency. Its basic idea is to determine resource allocation μ based on the preferences of the sampled students. Then we run DA based on μ. The entire mechanism is carefully designed to guarantee strategyproofness. The idea of dividing students/participants into two groups and utilizing the information obtained by one group to appropriately set parameters for the mechanism applied to another group is a popular technique to guarantee strategyproofness in auction domains [7, 15]. To the best of our knowledge, applying this idea in two-sided matching to develop a strategyproof mechanism is novel.

Mechanism 4 (Sample and Vote Deferred Acceptance (SVDA)).

Step 1: *Select $S' \subseteq S$, which we call the sampled students. 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.*

We use the following simple method to decide allocation $R \setminus R'$ based on the preferences of S'. For each r, each $s \in S'$ (hypothetically) votes for candidates T_r based on $\succ_s$, where each project obtains a Borda score based on $\succ_s$. Then r is allocated to the winner.

The details of this voting procedure do not affect SVDA's theoretical properties, e.g., whether a student can vote for a project to which she is unacceptable or not, or how ties are broken. Thus, they can be arbitrarily determined. An appropriate way for choosing sampled students is domain dependent. If we want "ex-ante fairness", we might choose sampled students uniformly at random. If there exists a distinguished class of students, e.g., scholarship students, they would be reasonable candidates as sampled students.

We show an example how SVDA works. Assume the setting in Example 2.2. In Step 1, assume $S' = \{s_1\}$, i.e., s_1 is the only sampled student. In SD, s_1 is matched to her first-choice project, p_1. In Step 2, the minimal allocation to make $\{(s_1, p_1)\}$ feasible is allocating r_1 to p_1. Thus, $R' = \{r_1\}$. Next the allocation of $R \setminus R' = \{r_2\}$ is determined by the preference of s_1. Then r_2 is allocated to p_4 based on T_{r_2}. In Step 3, since s_1 is fixed, the remaining capacities of p_1, p_2, p_3, and p_4 are 1, 0, 0, and 1. $S \setminus S' = \{s_2, s_3, s_4\}$ are matched by DA. Thus, s_2 is matched to p_4, s_3 is matched to p_1, and s_4 is unmatched. The result is not fair since s_2 has justified envy toward s_1. Nonwastefulness is not satisfied either since s_2 possibly claims an empty seat in p_3.

We show that SVDA satisfies several fundamental desiderata.

Theorem 4.4. *SVDA is strategyproof, resource efficient, weakly nonwasteful, fair among students in $S \setminus S'$, and no sampled student has justified envy toward another regular student.*

Proof. SVDA is clearly strategyproof for S' since SD is strategyproof [17]. SVDA is also strategyproof for $S \setminus S'$ since DA is strategyproof [9, 43] and the capacity of each project is determined exogenously for $S \setminus S'$.

Assume students unanimously prefer p over p'. When assigning S', any resource r such that $p, p' \in T_r$ is never allocated to p' since students apply to p before applying to p'. It also never wins in the voting procedure. Consequently, SVDA satisfies resource efficiency.

Assume student s is matched to p (which can be $\emptyset$). She applied to any project p', which is ranked higher than 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, SVDA satisfies weak nonwastefulness.

Concerning fairness, since DA is fair [13], no regular student has justified envy toward another regular student. Assume 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. □

Theorem 4.5. *SVDA 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 favorite 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 SVDA. □

Theorem 4.6. *SVDA is weakly group strategyproof.*

Proof. Since SD is group strategyproof [35], 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.

Furthermore, since DA is weakly group strategyproof [5] and project quotas are exogenously given by the preferences of 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 SVDA. □

Indeed, 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 doing so is not beneficial for her. Since weak group strategyproofness requires that *all members* must benefit, the existence of such a coalition does not contradict the fact that SVDA is weakly group strategyproof.

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

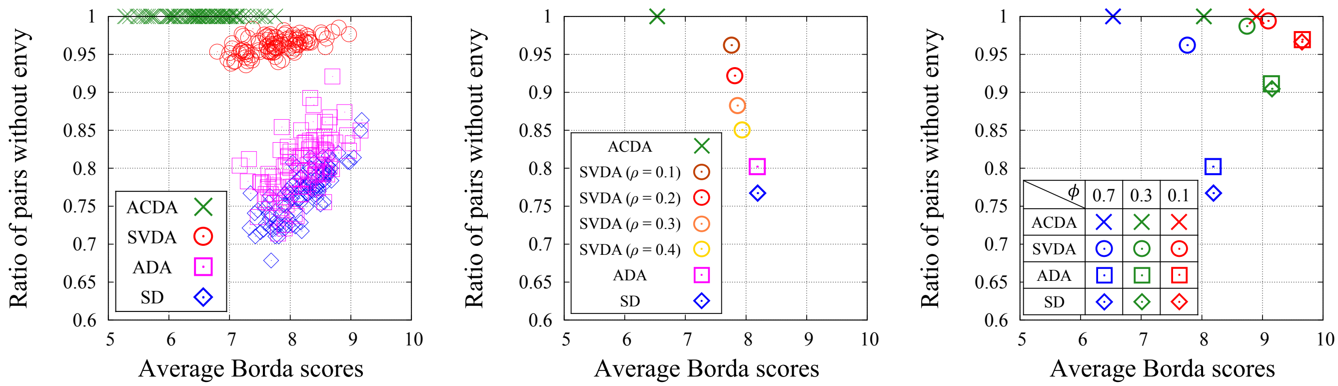
Theorem 4.7. *SVDA is Pareto efficient if all student preferences are identical and each project assumes that all students are acceptable.*

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. According to the votes of the sampled students, resource allocation μ is determined. In μ, any resource r 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. 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. Assigning a student in S_k to a better project is impossible without affecting the students in S' or $S_{k'}$ (where $k' < k$). Thus, no matching Pareto dominates the matching obtained by SVDA. □

Theorem 4.8. *SVDA is fair if all the projects have an identical preference and the sampled students are selected based on it.*

Proof. From Theorem 4.4, in SVDA, 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, but this 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 with $s \succ_{p'} s'$. □

SVDA 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



(a) Scatter plot ($\phi = 0.7$, $\rho = 0.1$) **(b) Effect of ρ ($\phi = 0.7$)** **(c) Effect of ϕ ($\rho = 0.1$)**

Figure 2: Trade-off between efficiency and fairness

amount of resources, which is sufficient for the demand of sampled students. Furthermore, state-of-the-art MIP solvers, e.g., Gurobi optimizer [18], can also handle fairly large-scale feasibility problems.

5 EXPERIMENTAL EVALUATION

We consider a market with $|S| = 200$ students, $|P| = 10$ projects, and $|R| = 20$ resources.[5] For each resource r, we randomly generate T_r such that each project p is included in T_r with probability 0.2. The capacity of each resource is $1, 5, 10, 15$, or 20 (the number of resources for each capacity is 4). Student preferences are generated with the Mallows model [8, 36, 37, 48]. 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 (uniformly randomly chosen from all possible preferences in our experiment), 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 uniform distribution (which is equivalent to impartial culture [14, 40] in our setting), and as ϕ increases, it quickly converges to the constant distribution that returns $\succ_{\widehat{s}}$. The preference of each project $\succ_p$ is drawn uniformly at random. We create 100 instances for each parameter setting and compare SVDA with other mechanisms. 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 Gurobi optimizer to solve Feasibility in SD and SVDA.

To illustrate the trade-off between efficiency and fairness, we plot the results of the obtained matching in a two-dimensional space in Figure 2, 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 northeast are preferable. For SVDA, we set the ratio of sampled students $\rho = |S'|/|S|$ to 0.1. Figure 2 (a) illustrates that SVDA strikes a good balance between efficiency and fairness. Each point represents the result of one instance for one mechanism. Figure 2 (b) shows the average for 100 problem instances for each mechanism. We vary ρ from 0.1 to 0.4. When it is small, SVDA resembles ACDA. By increasing ρ, it gradually resembles SD. Thus, by controlling parameter ρ, we can further fine-tune the balance. In Figure 2 (c), we vary spread parameter ϕ from 0.1 to 0.7. When it is large, the competition among students becomes more severe and resource allocation significantly affects their welfare. When ϕ is small, the difference among mechanisms becomes smaller. One might argue that SVDA works only when the sampled students resemble regular students. Although this is true to some extent, when student preferences are diverse, all the mechanisms work reasonably well. We also ran experiments with different voting procedures (simple majority and Copeland) and found quite similar results.

6 CONCLUSION AND FUTURE WORK

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. Then we developed a strategyproof mechanism called SVDA and proved that it is resource efficient, weakly nonwasteful, fair among some students, weakly Pareto efficient, and weakly group strategyproof. Finally, we numerically showed that it strikes a good balance between fairness and efficiency.

Our future works include theoretically identifying the optimal sample size and dealing with the case where various constraints are imposed on the allocation of resources, e.g., the total number of resources that can be allocated to each project is bounded.

ACKNOWLEDGMENT

This work is partially supported by JSPS KAKENHI Grant JP17H00761 and JST SICORP JPMJSC1607.

[5] Since SD and ADA are computationally expensive, running experiments for larger markets is time-consuming while SVDA (and ACDA) can handle much larger markets.

REFERENCES

[1] David J. Abraham, Robert W. Irving, and David F. Manlove. 2007. Two Algorithms for the Student-Project Allocation Problem. *Journal of Discrete Algorithms* 5, 1 (2007), 73–90.

[2] Ahmet Alkan. 1988. Nonexistence of Stable Threesome Matchings. *Mathematical Social Sciences* 16, 2 (1988), 207–209.

[3] Haris Aziz, Péter Biró, Tamás Fleiner, Serge Gaspers, Ronald de Haan, Nicholas Mattei, and Baharak Rastegari. 2017. Stable Matching with Uncertain Pairwise Preferences. In *Proceedings of the 16th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2017)*. 344–352.

[4] Haris Aziz, Serge Gaspers, Zhaohong Sun, and Toby Walsh. 2019. 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)*. 377–385.

[5] Salvador Barberà, Dolors Berga, and Bernardo Moreno. 2016. Group strategy-proofness in private good economies. *American Economic Review* 106, 4 (2016), 1073–99.

[6] Péter Biró, Tamás Fleiner, Robert W. Irving, and David F. Manlove. 2010. The College Admissions Problem with Lower and Common Quotas. *Theoretical Computer Science* 411, 34-36 (2010), 3136–3153.

[7] Christian Borgs, Jennifer Chayes, Nicole Immorlica, Mohammad Mahdian, and Amin Saberi. 2005. Multi-unit Auctions with Budget-constrained Bidders. In *Proceedings of the 6th ACM Conference on Electronic Commerce (EC 2005)*. 44–51.

[8] Joanna Drummond and Craig Boutilier. 2013. Elicitation and Approximately Stable Matching with Partial Preferences. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013)*. 97–105.

[9] Lester E Dubins and David A Freedman. 1981. Machiavelli and the Gale-Shapley Algorithm. *The American Mathematical Monthly* 88, 7 (1981), 485–494.

[10] Lars Ehlers, Isa E. Hafalir, M. Bumin Yenmez, and Muhammed A. Yildirim. 2014. School Choice with Controlled Choice Constraints: Hard Bounds versus Soft Bounds. *Journal of Economic Theory* 153 (2014), 648–683.

[11] Tamás Fleiner and Naoyuki Kamiyama. 2016. A Matroid Approach to Stable Matchings with Lower Quotas. *Mathematics of Operations Research* 41, 2 (2016), 734–744.

[12] Daniel Fragiadakis, Atsushi Iwasaki, Peter Troyan, Suguru Ueda, and Makoto Yokoo. 2016. Strategyproof Matching with Minimum Quotas. *ACM Transactions on Economics and Computation* 4, 1 (2016), 6:1–6:40.

[13] David Gale and Lloyd Stowell Shapley. 1962. College Admissions and the Stability of Marriage. *The American Mathematical Monthly* 69, 1 (1962), 9–15.

[14] William V Gehrlein and Peter C Fishburn. 1976. The probability of the paradox of voting: A computable solution. *Journal of Economic Theory* 13, 1 (1976), 14–25.

[15] Andrew V. Goldberg, Jason D. Hartline, and Andrew Wright. 2001. Competitive auctions and digital goods. In *Proceedings of the 12th Annual Symposium on Discrete Algorithms (SODA 2001)*. 735–744.

[16] Masahiro Goto, Atsushi Iwasaki, Yujiro Kawasaki, Ryoji Kurata, Yosuke Yasuda, and Makoto Yokoo. 2016. Strategyproof Matching with Regional Minimum and Maximum Quotas. *Artificial Intelligence* 235 (2016), 40–57.

[17] Masahiro Goto, Fuhiko Kojima, Ryoji Kurata, Akihisa Tamura, and Makoto Yokoo. 2017. Designing Matching Mechanisms under General Distributional Constraints. *American Economic Journal: Microeconomics* 9, 2 (2017), 226–262.

[18] Gurobi. 2019. Gurobi Optimization. (2019). http://www.gurobi.com/.

[19] Isa E. Hafalir, M. Bumin Yenmez, and Muhammed A. Yildirim. 2013. Effective Affirmative Action in School Choice. *Theoretical Economics* 8, 2 (2013), 325–363.

[20] Koki Hamada, Kazuo Iwama, and Shuichi Miyazaki. 2016. The Hospitals/Residents Problem with Lower Quotas. *Algorithmica* 74, 1 (2016), 440–465.

[21] Naoto Hamada, Chia-Ling Hsu, Ryoji Kurata, Takamasa Suzuki, Suguru Ueda, and Makoto Yokoo. 2017. Strategy-proof School Choice Mechanisms with Minimum Quotas and Initial Endowments. *Artificial Intelligence* 249 (2017), 47–71.

[22] John William Hatfield and Paul R. Milgrom. 2005. Matching with Contracts. *American Economic Review* 95, 4 (2005), 913–935.

[23] Hadi Hosseini, Kate Larson, and Robin Cohen. 2015. On Manipulablity of Random Serial Dictatorship in Sequential Matching with Dynamic Preferences. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI 2015)*. 4168–4169.

[24] Chien-Chung Huang. 2007. 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)*. 558–569.

[25] Anisse Ismaili, Naoto Hamada, Yuzhe Zhang, Takamasa Suzuki, and Makoto Yokoo. 2019. Weighted Matching Markets with Budget Constraints. *Journal of Artificial Intelligence Research* 65 (2019), 393–421.

[26] Anisse Ismaili, Kentaro Yahiro, Tomoaki Yamaguchi, and Makoto Yokoo. 2019. Student-Project-Resource Matching-Allocation Problems: Two-Sided Matching Meets Resource Allocation. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2019)*. 2033–2035.

[27] Anisse Ismaili, Tomoaki Yamaguchi, and Makoto Yokoo. 2018. 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)*. 226–241.

[28] Yuichiro Kamada and Fuhito Kojima. 2015. Efficient Matching under Distributional Constraints: Theory and Applications. *American Economic Review* 105, 1 (2015), 67–99.

[29] Richard M. Karp. 1972. *Reducibility among Combinatorial Problems*. Springer, 85–103.

[30] Yasushi Kawase and Atsushi Iwasaki. 2017. Near-Feasible Stable Matchings with Budget Constraints. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI 2017)*. 242–248.

[31] Fuhito Kojima. 2012. School Choice: Impossibilities for Affirmative Action. *Games and Economic Behavior* 75, 2 (2012), 685–693.

[32] Fuhito Kojima, Akihisa Tamura, and Makoto Yokoo. 2018. Designing Matching Mechanisms Under Constraints: An Approach from Discrete Convex Analysis. *Journal of Economic Theory* 176 (2018), 803–833.

[33] Bernhard Korte and Jens Vygen. 2018. *Combinatorial Optimization: Theory and Algorithms*. Springer.

[34] Ryoji Kurata, Naoto Hamada, Atsushi Iwasaki, and Makoto Yokoo. 2017. Controlled School Choice with Soft Bounds and Overlapping Types. *Journal of Artificial Intelligence Research* 58 (2017), 153–184.

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

[36] Tyler Lu and Craig Boutilier. 2014. Effective Sampling and Learning for Mallows Models with Pairwise-Preference Data. *Journal of Machine Learning Research* 15 (2014), 3963–4009.

[37] Colin L Mallows. 1957. Non-null ranking models. I. *Biometrika* 44, 1-2 (06 1957), 114–130.

[38] David F. Manlove, Duncan Milne, and Sofiat Olaosebikan. 2018. An Integer Programming Approach to the Student-Project Allocation Problem with Preferences over Projects. In *Proceedings of the International Symposium on Combinatorial Optimization (ISCO 2018)*. 313–325.

[39] David F. Manlove and Gregg O'Malley. 2008. Student-Project Allocation with preferences over Projects. *Journal of Discrete Algorithms* 6, 4 (2008), 553 – 560.

[40] Nicholas Mattei and Toby Walsh. 2013. PrefLib: A Library for Preferences http://www.preflib.org. In *Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT 2013)*. 259–270.

[41] Cheng Ng and Daniel S. Hirschberg. 1991. Three-Dimensional Stable Matching Problems. *SIAM Journal on Discrete Mathematics* 4, 2 (1991), 245–252.

[42] Yasunori Okumura. 2019. School Choice with General Constraints: A Market Design Approach for the Nursery School Waiting List Problem in Japan. *The Japanese Economic Review* (2019).

[43] Alvin E Roth. 1982. The Economics of Matching: Stability and Incentives. *Mathematics of Operations Research* 7, 4 (1982), 617–628.

[44] Alvin E. Roth and Marilda A. Oliveira Sotomayor. 1990. *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis (Econometric Society Monographs)*. Cambridge University Press.

[45] Tayfun Sönmez. 2013. Bidding for Army Career Specialties: Improving the ROTC Branching Mechanism. *Journal of Political Economy* 121, 1 (2013), 186–219.

[46] Tayfun Sönmez and Tobias B. Switzer. 2013. Matching with (Branch-of-Choice) Contracts at the United States Military Academy. *Econometrica* 81, 2 (2013), 451–488.

[47] Kentaro Tomoeda. 2018. Finding a stable matching under type-specific minimum quotas. *Journal of Economic Theory* 176 (2018), 81–117.

[48] Jack D. Tubbs. 1992. Distance Based Binary Matching. In *Computing Science and Statistics*. Springer, 548–550.

[49] Kentaro Yahiro, Yuzhe Zhang, Nathanaël Barrot, and Makoto Yokoo. 2018. Strategyproof and Fair Matching Mechanism for Ratio Constraints. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2018)*. 59–67.

[50] Yuzhe Zhang, Kentaro Yahiro, Nathanaël Barrot, and Makoto Yokoo. 2018. Strategyproof and Fair Matching Mechanism for Union of Symmetric M-convex Constraints. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI 2018)*. 590–596.