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:

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.,...
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 $(S, P, R, >_S, >_P, T_R, q_R)$.

- $S = \{s_1, \ldots, s_{|S|}\}$ 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.
- $>_S = (>_s)_{s \in S}$ are the student strict preferences over set $S \cup \{\emptyset\}$. Symbol $\emptyset$ means that a student is not assigned to any project.
- $>_P = (>_p)_{p \in P}$ are the project strict preferences over set $S \cup \{\emptyset\}$. Symbol $\emptyset$ means that a project is assigned to 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:

- $s_1: p_1 >_P p_2 >_P p_3 >_P p_4 >_S s_1 >_S s_2 >_S s_3 >_S s_4 >_S \emptyset$.
- $s_2: p_4 >_P p_3 >_P p_2 >_P p_1 >_S s_1 >_S s_2 >_S s_3 >_S s_4 >_S \emptyset$.
- $s_3: p_1 >_P p_2 >_P p_3 >_P p_4 >_S s_1 >_S s_2 >_S s_3 >_S s_4 >_S \emptyset$.
- $s_4: p_4 >_P p_3 >_P p_2 >_P \emptyset >_P p_1 >_S s_1 >_S s_2 >_S s_3 >_S s_4 >_S \emptyset$.

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: $(2, 0, 1, 0), (2, 0, 0, 1), (0, 2, 1, 0), (0, 2, 0, 1)$.

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 $p >_S s \emptyset$ holds (resp. $s >_P \emptyset$). Let $X$ denote the set of all contracts that are acceptable to the projects.

$\begin{array}{c}
\text{A matching is a set of contracts which satisfy the following conditions.} \\
\text{Definition 2.3 (Matching). A matching is a subset } Y \subseteq X, \text{ where for every student } s \in S, Y_s = \{(s, p) \mid (s, p) \in X\}, \text{ either } |Y_s| = 0, \text{ or } Y_s = \{(s, p)\} \text{ and } p >_S s \emptyset \text{ hold.}
\end{array}$

Footnote: 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.
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 feasibility of a matching is defined based on this description.

**Definition 2.4 (Allocation).** An allocation $\mu : R \rightarrow P$ maps each resource $r$ to a project $\mu(r) \in T_r$. Let $q_\mu(p) = \sum_{r \in \mu^{-1}(p)} q_r$.

**Definition 2.5 (Feasibility).** A feasible matching $(Y, \mu)$ is a matching-allocation pair where $|Y(p)| \leq q_\mu(p)$ 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 $\mu$).** For feasible matching $(Y, \mu)$, student $s$ claims an empty seat in project $p$ with $\mu$ if the following conditions hold:

- $p >_s Y(s)$, and
- $Y \n Y(s, Y(s)) \cup \{(s, p)\}$ is feasible with $\mu$.

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

**Definition 2.7 (Nonwastefulness).** For feasible matching $(Y, \mu)$, student $s$ possibly claims an empty seat in project $p$ with $\mu'$ if $\exists s' >_s Y(s)$ such that $s$ claims an empty seat in $p$ with $\mu'$. Feasible matching $(Y, \mu)$ 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 $\mu'$. Note that $\mu'$ can be different from $\mu$. Thus, $s$ can possibly claim an empty seat in $p$ even if it is impossible to move her to $p$ with current allocation $\mu$ as long as it becomes possible with a different and better allocation $\mu'$. 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.

![Figure 1: SPR instance: matching $\hat{Y}$ and allocation $\hat{\mu}$ in Example 2.2](image)

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 $(Y, \mu)$, student $s$ has justified envy toward student $s'$ if for project $p$ such that $s' \in Y(p)$, $p >_s Y(s)$ and $s >_p s'$ hold. A feasible matching $(Y, \mu)$ 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 $(Y, \mu)$ 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 $Y'(s) >_s Y(s)$ 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 $\mu'$. 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
s1 and s2 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 mecha-

anism outputs a feasible matching \( (Y, \mu) \). 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 strat-
egyproof 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, s1, s2,
two projects, p1, p2, and a unitary resource compatible with both.
The student preferences are \( p_1 > s_1 \) and \( p_2 > s_1 \). The project
preferences are \( s_2 > p_1, s_1 > p_1, s_2 > p_2 \). 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 condi-
tions on efficiency.

Definition 3.2 (Weak Nonwastefulness). Feasible matching \( (Y, \mu) \)
is weakly nonwasteful if no student claims an empty seat with
current allocation \( \mu \).

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 match-
ing \( (Y, \mu) \), student \( s \) strongly claims an empty seat if \( Y(s) = \emptyset \), and
\( \forall \mu' \), such that \( (Y, \mu') \) is feasible, \( \exists \mu \) in which \( s \) claims an empty
seat with \( \mu' \). 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
\( \mu' \), project \( p \) exists such that \( s \) claims an empty seat in \( p \) with \( \mu' \).
Note that \( p \) can be different for each \( \mu' \).

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 un-
assigned and claims an empty seat in project \( p \) with current allocation
\( \mu \). 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 de-

fine unanimous preferences.

Definition 3.4 (Unanimous Preference). Students unanimously prefer
\( p \) over \( p' \) if for every \( s \in S \), \((s, p) \in X \) and \( p > s \) \( p' \) 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 \( \mu \) is re-
source 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 theo-
rem shows that Pareto efficiency also implies resource efficiency.

Theorem 3.6. If feasible matching \( (Y, \mu) \) is Pareto efficient, then
allocation \( \mu' \) exists such that \( (Y, \mu') \) is feasible and \( \mu' \) is resource
efficient.

Proof. For contradiction, assume \( Y \) is Pareto efficient, and all
students unanimously prefer \( p \) over \( p' \), but for any \( s \) such that \( (Y, \mu) \)
is feasible, resource \( r \) is allocated to \( p' \) while \( p, p' \in T_r \) holds. Con-
sider \( \mu' \) obtained from \( \mu \), such that \( r \) is re-assigned to \( p \). If \( (Y, \mu') \)

is feasible, we repeat the same procedure. \( (Y, \mu') \) eventually becomes
infeasible (otherwise, we obtain resource efficient \( \mu' \), which contra-
dicts 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 \( (Y', \mu') \) 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 non-
wa isteful, resource efficient, and strategyproof.

Proof. Consider the following situation: three students, s1, s2, s3,
three projects, p1, p2, p3, one resource, r, with \( q_r = 2 \), and \( T_r = \{(p_1, p_2, p_3)\} \). The following are the preferences:
\[
\begin{align*}
    s_1: p_2 &> p_3 > p_1 > \emptyset, \\
    s_2: p_3 &> p_1 > p_2 > \emptyset, \\
    s_3: p_1 &> p_2 > p_3 > \emptyset, \\
\end{align*}
\]
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 s1 and
s2 to p1, allocating s2 and s3 to p2, or allocating s3 and s1 to p3. From
the symmetry, we can assume r is allocated to p1 and s1 and s2 are
assigned to p1 w.l.o.g. Next we examine the case where preference
of s3 is changed to \( p_3 > p_1 > p_2 > \emptyset \). From resource efficiency,
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 $\mu$ exists such that $(Y, \mu)$ is feasible), then any of its subsets $Y' \subseteq Y$ is also feasible with some allocation $\mu'$. SPR clearly satisfies heredity: if $(Y, \mu)$ is feasible, for any $Y' \subseteq Y$, $(Y, \mu)$ 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 $\mu$ exist such that $(Y, \mu)$ is feasible?

We settle its computational complexity by the reduction from a partition problem, which is known to be $\mathsf{NP}$-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 $\mathsf{NP}$-complete.

Proof. For yes instances, whether $(Y, \mu)$ is feasible can be verified in polynomial time when $\mu$ is given as a certificate. Hence, this problem belongs to class $\mathsf{NP}$.

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 $t$ resources $R = \{r_1, \ldots, r_t\}$. 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 Feasibility is yes, and vice versa.

Since Partition is $\mathsf{NP}$-complete, Feasibility is also $\mathsf{NP}$-complete.

To verify feasibility, we need to solve a Mixed Integer Programming (MIP) instance. For a special case where $q_{r_i} = 1$ for all $r_i$ and $T_r$ has a laminar structure, the distributional constraints form an $\mathbb{M}^2$-convex set; a generalized mechanism can obtain a fair matching [32] since Feasibility is no longer $\mathsf{NP}$-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 $Y$ denote all the possible matchings, each of which can be feasible with some allocation. The first student, $s_1$, chooses subset $Y_1 \subseteq Y$, such that she equally prefers any matching in $Y_1$ and strictly prefers any matching in $Y \setminus Y_1$. In other words, she chooses her most preferred matchings in $Y$. 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 $Y$. In the setting of Example 2.2, $s_1$ will choose $Y_1 = \{Y \in Y \mid (s_1, p_1) \in Y\}$, i.e., all elements in $Y$ such that $s_1$ is allocated to $p_1$. Then next student $s_2$ chooses $Y_2 \subseteq Y_1$ in a similar way; she chooses her most preferred matchings within $Y_1$, and so forth. In the setting of Example 2.2, $s_2$ will choose $Y_2 = \{Y \in Y_1 \mid (s_2, p_1) \in Y\}$, i.e., all elements in $Y_1$ 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)).

\begin{enumerate}
\item $Y \leftarrow \emptyset, k \leftarrow 1$.
\item 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 $\mu'$. $Y \leftarrow Y \cup \{(s_k, p)\}$ (if no such $p$ exists, $s_k$ is not assigned to any project).
\item $k \leftarrow k + 1$. Go to Step 2.
\end{enumerate}

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 $\mu$ 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)).

\begin{enumerate}
\item Choose $\mu$ independently from $\rightarrow$.
\item Run the standard DA, assuming the maximum quota of each project $p$ is $\sum_{r \in W(p)} q_r$ and obtain matching $Y$.
\item Return $(Y, \mu)$.
\end{enumerate}
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 \( \sum_{r \mid T_r \supseteq p} q_r \), 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 \( \mu \) exists with which \( Y \cup \{ s(p), s(q) \} \) 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 \( L \leftarrow (s_1, s_2, \ldots) \), \( q_p^t \leftarrow q_p \) 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 \( q_p^{k-1} \) for each \( p \in P \).

(i) If all students in \( L \) are already selected, then output \( Y' \cup Y' \) and terminate the mechanism.

(ii) If no project \( p_1 \) 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 \( q_p - |Y'(p)| \). 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 SD 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 \( p_1 \) using DA. Then, project \( p_2 \) 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.

**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 \( \mu \) based on the preferences of the sampled students. Then we run DA based on \( \mu \). 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 \( R'' \subseteq R' \) 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 \( \mu \) 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 \( >_s \), where each project obtains a Borda score based on \( >_s \). Then \( r \) is allocated to the winner.

<table>
<thead>
<tr>
<th>Mechanism</th>
<th>Fairness</th>
<th>PE</th>
<th>NW</th>
<th>Weak NW</th>
<th>RE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD [17]</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>ADA [17]</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>SVDA</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>ACDA [17]</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
</tbody>
</table>
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 \( \mu' \) exists such that \( s \) can be assigned to \( p' \). If \( s \in S \setminus S' \), \( s \) cannot be assigned to \( p' \) with current allocation \( \mu \). 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 \( p_1 \succ s, p_2 \succ s, \ldots \). Since SD is Pareto efficient, it is impossible to assign sampled student \( s \) to a better project without disadvantage another sampled student \( s' \) who was assigned before \( s \). According to the votes of the sampled students, resource allocation \( \mu \) is determined. In \( \mu \), any resource \( r \) is allocated to \( p_1 \) 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., \( s \succ s', s \) and \( p' \succ p \) hold). For case (i), \( s' \succ p' \) must hold from the assumption that every project unanimously prefers a sampled student over a regular student, but this contradicts our assumption that \( s \succ p' \). For case (ii), \( s' \) must be assigned before \( s \), which means every project unanimously prefers \( s' \) over \( s \). This contradicts with \( s \succ p' \).

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
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. 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 $Pr(\succ_s)$:

$$Pr(\succ_s) = \frac{\exp(-\phi \cdot d(\succ_s, \succ_s^c))}{\sum_{s' \in S} \exp(-\phi \cdot d(\succ_s, \succ_{s'}^c))}.$$ 

Here $\phi \in \mathbb{R}$ denotes a spread parameter, $\succ_s^c$ is a central preference (uniformly randomly chosen from all possible preferences in our experiment), and $d(\succ_s, \succ_s^c)$ represents the Kendall tau distance, which is the number of pairwise inversions between $\succ_s$ and $\succ_s^c$. In short, student preferences are distributed around a central preference with spread parameter $\phi$. When $\phi = 0$, the Mallows model becomes identical to uniform distribution (which is equivalent to impartial culture [14, 40] in our setting), and as $\phi$ increases, it quickly converges to the constant distribution that returns $\succ_s^c$. 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 $|P|i - i + 1$, 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 works well in balancing 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 $\rho$ from 0.1 to 0.4. When it is small, SVDA resembles ACDA. By increasing $\rho$, it gradually resembles SD. Thus, by controlling parameter $\rho$, we can further fine-tune the balance. In Figure 2 (c), we vary spread parameter $\phi$ 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 $\phi$ 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 non-wasteful, 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.
REFERENCES


