Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Assignment Mechanisms Under Distributional Constraints

Itai Ashlagi, Amin Saberi, Ali Shameli

To cite this article:
Itai Ashlagi, Amin Saberi, Ali Shameli (2020) Assignment Mechanisms Under Distributional Constraints. Operations Research 68(2):467-479. https://doi.org/10.1287/opre.2019.1887

informs
http://pubsonline.informs.org/journal/opre

OPERATIONS RESEARCH
Vol. 68, No. 2, March–April 2020, pp. 467–479
ISSN 0030-364X (print), ISSN 1526-5463 (online)

Crosscutting Areas

Assignment Mechanisms Under Distributional Constraints

Itai Ashlagi,[a] **Amin Saberi,**[a] **Ali Shameli**[a]

[a] Department of Management Science and Engineering, Stanford University, Stanford, California 94305
Contact: iashlagi@stanford.edu (IA); saberi@stanford.edu (AmS); shameli@stanford.edu, https://orcid.org/0000-0003-4246-4279 (AlS)

Received: May 18, 2017
Revised: July 18, 2018; January 4, 2019; March 15, 2019
Accepted: May 14, 2019
Published Online in Articles in Advance: February 27, 2020

Subject Classifications: analysis of algorithms
Area of Review: Games, Information, and Networks

https://doi.org/10.1287/opre.2019.1887

Abstract. We generalize the serial dictatorship (SD) and probabilistic serial (PS) mechanism for assigning indivisible objects (seats in a school) to agents (students) to accommodate distributional constraints. Such constraints are motivated by equity considerations. Our generalization of SD maintains several of its desirable properties, including strategyproofness, Pareto optimality, and computational tractability, while satisfying the distributional constraints with a small error. Our generalization of the PS mechanism finds an ordinally efficient and envy-free assignment while satisfying the distributional constraint with a small error. We show, however, that no ordinally efficient and envy-free mechanism is also weakly strategyproof. Both of our algorithms assign at least the same number of students as the optimum fractional assignment.

Funding: I. Ashlagi was supported by the National Science Foundation (NSF) [Grant SES-1254768] and the U.S.-Israel Binational Science Foundation [Grant 2012344]. A. Saberi's research is supported by the National Science Foundation (NSF) [Grant 1812919] and by the Office of Naval Research (NSF) [Grant N000141612893].

Keywords: market design • assignment • matching • refugee • strategyproof

1. Introduction

We consider the problem of assigning indivisible objects to agents with privately known preferences who are interested in consuming at most one object. One classic solution for this problem is the serial dictatorship (SD) mechanism, which considers agents in a certain order and assigns to each agent her most preferred object from the remaining objects. Some applications of SD (with some variations) include school choice (Abdulkadiroğlu et al. 2009, Pathak and Sönmez 2013), college admissions (Chen and Kesten 2017, Baswana et al. 2019), on-campus housing, and office allocation (Chen and Snmez 2002). The SD mechanism has several desirable properties. It is strategyproof, Pareto efficient, and computationally efficient. Random serial dictatorship (RSD), which picks the order of agents uniformly at random, also treats agents with identical preferences equally in the sense that it assigns them each object with the same probability.

Despite RSD's attractive features it entails an unambiguous efficiency loss ex ante. Specifically, it fails to be ordinally efficient, because agents may be better off trading probability shares before the outcome is realized (for a formal definition see Section 4). The probabilistic serial (PS) mechanism, introduced by Bogomolnaia and Moulin (2001), eliminates the inefficiency present in RSD. The PS mechanism can be described as follows. Imagining that each object is divisible, all agents simultaneously "eat" at rate one from their most preferred object among the remaining objects. This process continues for one unit of time, after which objects are assigned randomly to agents with probabilities that correspond to the divisible shares each agent has consumed. Note that these mechanisms can be simulated using a computer after agents report their ordinal preferences over objects. One drawback of PS is that it is not strategyproof but satisfies a weaker notion, which is known as weak strategyproofness (Bogomolnaia and Moulin 2001).

The goal of this paper is to extend the SD and PS mechanisms to settings with distributional constraints. Constraints of this sort arise in various contexts. Some school districts impose quotas for students based on geographic location in order to increase socioeconomic integration. Regional quotas are imposed in resident matching in Japan so that programs in rural areas do not remain underassigned (Kamada and Kojima 2014). Quotas are imposed when assigning cadets to army branches (Sönmez and Switzer 2013). Similar policies are adopted in college admissions in various countries (Biró et al. 2010, Braun et al. 2014). When assigning refugees who often have family needs, various constraints arise owing to local service capacities (Delacrétaz et al. 2016). In each of these applications, it is desirable to leave as few unassigned agents as possible.

One challenge with generalizing these mechanisms to accommodate distributional constraints is computational; even checking whether there is an assignment that satisfies the distributional constraints is NP-complete (Ashlagi et al. 2019). Given this, we treat the distributional constraints as soft and look for assignments that do not violate the given constraints by much, as following Nguyen and Vohra (2017). In some of the above applications it is arguably reasonable to allow for small violations of these constraints.

Next we describe our contributions. For exposition purposes, we formulate the problem in terms of assigning students to schools. Each student in the model has a publicly known type (a type can encode, for example, the neighborhood of the student, socioeconomic status, or race) and a privately known ordinal ranking over schools. Each school has lower- and upper-bound quotas for the number of students of certain types that it can admit. We are interested in assigning as many students to schools as possible. We refer to this property as "allocative efficiency."

Our main contribution is a generalization of the serial dictatorship mechanism, which maintains strategyproofness, Pareto efficiency, and computational efficiency. It also produces assignments that violate the distributional constraints by no more than the number of available types.

In addition to the above properties, the number of students our mechanism assigns is at least the number of students that can be fractionally assigned, subject to distributional constraints (see, e.g., Ehlers et al. 2014 and Kamada and Kojima 2014). We refer to this benchmark as OPT. Observe that some constraints may need to be violated to achieve this benchmark. One assumption we make is that all schools are acceptable to all students; this is important to achieve allocative efficiency together with strategyproofness (thus preventing students from truncating their preference lists). This is a reasonable assumption when outside options are very limited, as arguably the case when assigning refugees, assigning cadets to military schools, or assigning students with few local private schools.

The key idea behind the mechanism is to carefully design a menu of schools that are available for each student who is about to be assigned. This is done by iteratively solving a set of linear programs before each assignment, one for each school, which checks whether the student can possibly be assigned to the school in a way that eventually at least OPT students will be assigned.

We further introduce a generalization of the probabilistic serial algorithm, which produces an ordinally efficient assignment. The main idea is that during the "eating" process, a student who is about to violate a constraint that is associated with the school she is eating from switches to eat from her next most preferred school. The eating process terminates with a fractional assignment, which is then implemented as a lottery over integral assignments such that no distributional constraint is violated by more than the number of existing types. We further show that the generalized PS is envy-free (within-type) and ordinally efficient. However, in contrast to the setting without distributional constraints, we show that no mechanism is envy-free, ordinally efficient, and weakly strategyproof.

Finally, we note that violating constraints is necessary for a couple of reasons. First, we wish to assign at least OPT students. Second, the set of constraints is more general than the bihierarchical structure that is necessary and sufficient for implementing a random assignment using a lottery over feasible assignments (Budish et al. 2013). In a related work Akbarpour and Nikzad (2019) consider a class of more general constraints and show how a given feasible fractional assignment can be implemented using lotteries over integral assignments with small errors (and do not consider the mechanism design question).

1.1. Related Work

There is a growing literature on assignment and matching mechanisms subject to distributional constraints. Several papers study which constraints allow implementation of affirmative action (Huang 2010, Fleiner and Kamiyama 2012, Kojima 2012, Hafalir et al. 2013, Kominers and Sönmez 2013, Westkamp 2013, Braun et al. 2014, Ehlers et al. 2014, Yokoi 2016). These papers consider either lower- or upper-bound constraints. Because satisfying lower-bound constraints is generally impossible (Biró et al. 2010), some of these studies also consider soft constraints, but without providing guarantees on constraint violations. Nor do these studies consider allocative efficiency.

More general constraints, like "regional-caps," have been considered (Kamada and Kojima 2014, 2017), and several studies have considered lower- and upper-bound constraints simultaneously (Ehlers et al. 2014, Goto et al. 2015, Fragiadakis et al. 2016, Hamada et al. 2016, Fragiadakis and Troyan 2017). These studies focus on constrained efficiency or weak stability and seek nonwasteful outcomes. Our paper allows for constraints on subsets of types while also seeking allocative efficiency. We focus, however, only on assignment problems with no priorities.

This paper is inspired by Nguyen and Vohra (2019) and Nguyen and Vohra (2017), which, respectively, study two-sided markets under complementarities and proportionality constraints and find mechanisms that implement stable matchings without violating each constraint by much. To bound the constraint violations, they adopt a novel approach using Scarf's Lemma. We build on more straightforward techniques, though based as well on linear programming. Moreover, we have a different objective, namely leaving few students unassigned. Noda (2018) has the same objective as in our paper. However, he assumes constraints cannot be violated and develops, under large market assumptions, strategyproof mechanisms that are only approximately optimal.

Finally, our paper assumes schools' preferences are given via ranking lists and quotas are given exogenously. We refer to Echenique and Yenmez (2015), who characterize schools' choice rules that account for diversity preferences and find that natural axioms yield such quotas.

2. Model

A school choice problem consists of a set of students $N = \{1, \dots, n\}$ and a set of schools $M = \{1, \dots, m\} \cup \{\phi\}$, where ϕ is an outside option. Every other school is referred to as a *regular* school.

Each student $i \in N$ has a *strict preference ordering* $\succ_i$ over M. We assume that all students prefer every regular school to the outside option, and later we discuss the robustness of the results based on this assumption. Each student $i \in N$ is associated with a commonly known type t_i, which belongs to a finite set of types denoted by T. Denote by C_t the number of students of type $t \in T$.

An *assignment* of students to schools is given by a matrix $[(x_{i,s})_{i\in N, s\in M}]$, where $x_{i,s}$ is the probability that i is assigned to s, and for all $i \in N$, $\sum_{s\in M} x_{i,s} = 1$. An assignment is *integral* if every student $i \in N$ is assigned to a single school $s \in M$ with probability 1. We refer to an integral assignment also as an *allocation*.

It will be useful to consider the assignment of students on the basis of their types. A vector $x = [(x_{t,s})_{t\in T, s\in M}]$ is called a *type-assignment* if for every type $t \in T$, $\sum_{s\in M} x_{t,s} = C_t$. Note that every assignment corresponds to a unique type-assignment. Throughout the paper, we will refer to a type-assignment simply as an assignment.

Next we introduce the distributional constraints. For every $s \in M$, let $Z(s) \subseteq 2^T$ be a collection of subsets $R \subseteq T$. For every $s \in M$ and every $R \in Z(s)$, we are given lower- and upper-bound *quotas* $\underline{q}_{R,s}$ and $\bar{q}_{R,s}$, respectively. ($\bar{q}_{T,s}$ can be thought of as the capacity of school s.) We assume there are no constraints imposed on ϕ, that is, $Z(\phi) = \emptyset$. Let $\underline{q} = [(\underline{q}_{R,s})_{s\in M, R\in Z(s)}]$ and $\bar{q} = [(\bar{q}_{R,s})_{s\in M, R\in Z(s)}]$. We refer to $\mathbf{q} = [\underline{q}, \bar{q}]$ as the *distributional constraints*.

We say that an assignment x is *feasible* with respect to $\mathbf{q}$ if

$$\underline{q}_{R,s} \le \sum_{t\in R} x_{t,s} \le \bar{q}_{R,s} \quad \forall s \in M, R \in Z(s).$$

An allocation x *Pareto dominates* another allocation y if no student is worse off in x than in y and at least one student is better off. An allocation x is said to be *Pareto efficient* with respect to $\mathbf{q}$ if there is no other allocation that is feasible with respect to $\mathbf{q}$ that Pareto dominates x.

Pareto-efficient allocations can differ with respect to the number of students assigned to regular schools (see Example 1 below). We are interested in maximizing the number of students that are assigned to regular schools. Consider the linear program (LP1), which attains this objective over all feasible fractional assignments and denote its objective by OPT. That is, OPT is the maximum (fractional) number of students that can be assigned to regular schools without violating the distributional constraints.

We say that x is *allocative efficient* if it assigns at least OPT many students to regular schools. We are interested in finding allocations that are allocative efficient, while violating each lower- and upper-bound quota by at most $|T|$.

$$\begin{aligned} \text{OPT} = \max_{x\in\mathbb{R}^{T\times M}} \quad & \sum_{t\in T} \sum_{s\in M\setminus\{\phi\}} x_{t,s} & & \\ \text{s.t.} \quad & \sum_{t\in R} x_{t,s} \le \bar{q}_{R,s}, & & s \in M, R \in Z(s) \\ & \sum_{t\in R} x_{t,s} \ge \underline{q}_{R,s}, & & s \in M, R \in Z(s) \qquad \text{(LP1)} \\ & \sum_{s\in M} x_{t,s} = C_t, & & t \in T \\ & x_{t,s} \ge 0, & & t \in T, s \in M. \end{aligned}$$

Throughout the paper we assume that (LP1) has a feasible solution.

Example 1 (Few Students Assigned To Regular Schools). This example illustrates that Pareto efficiency does not imply allocative efficiency and a Pareto-efficient assignment can result in many unassigned students. To see this, suppose there are three types of students, t_1, t_2, t_3, and one regular school s with 20 seats. We are given two constraints: (i) at most 10 students of types t_1 or t_2 can be assigned to s, and (ii) at most 10 students of types t_1 or t_3 can be assigned to s. Observe that assigning 10 students of type t_1 to s, or assigning 10 students of type t_2 and 10 students of type t_3 to s leads to Pareto efficiency.

Finally, a *mechanism* maps preference profiles to allocations. A mechanism is *strategyproof* if it is a weakly dominant strategy for every student to reveal her true preferences in the game induced by the mechanism.

3. Serial Dictatorship with Dynamic Menus

We present a generalization of SD for assignments with distributional constraints. The algorithm outputs an assignment that satisfies allocative efficiency and violates every lower- and upper-bound quota by at most $|T|$. As in SD, students in our algorithm are sequentially assigned to their most preferred school from a given menu. A key difference is that the menu given to every student is computed dynamically with the aid of a linear program.

Throughout the algorithm we maintain a vector $y = [(y_{t,s})_{t\in T, s\in M}]$ that keeps track of the (possibly fractional) quantity of students of type t assigned to school s. We refer to $\boldsymbol{y}$ as an incomplete assignment. We also maintain a vector $\boldsymbol{\Delta} = [(\Delta_{t,s})_{t\in T,s\in M}]$ to keep track of how much the lower- and upper-bound quotas corresponding to each type $t \in T$ and school $s \in M$ have changed so far. For the sake of convenience, define, for any school $s \in M$ and subset of types $R \in Z(s)$, $\boldsymbol{y}_{R,s} = \sum_{t\in R} \boldsymbol{y}_{t,s}$ and $\boldsymbol{\Delta}_{R,s} = \sum_{t\in R} \boldsymbol{\Delta}_{t,s}$.

To design each student's menu, we will need to solve the following auxiliary linear program (LP2). The linear program takes an incomplete assignment $\boldsymbol{y}$ and distributional constraints $[\underline{\mathbf{q}} + \boldsymbol{\Delta}, \bar{\mathbf{q}} + \boldsymbol{\Delta}]$ as input and looks for a feasible solution $\boldsymbol{x}$ that assigns the remaining students of each type in a way that is both allocative efficient and feasible with respect to the distributional constraints. The objective of (LP2) is to find such a solution that maximizes the quantity of students of a given type $\hat{t}$ that are assigned to a given school $\hat{s}$, which we denote by $f(\hat{t},\hat{s})$.

$$
\begin{aligned}
f(\hat{t},\hat{s}) = \max_{\boldsymbol{x}\in\mathbb{R}^{T\times M}} \ & \boldsymbol{x}_{\hat{t},\hat{s}} \\
\text{s.t.} \ & \sum_{t\in T}\sum_{s\in M\setminus\{\phi\}} \boldsymbol{x}_{t,s} + \sum_{s\in M\setminus\{\phi\}} \boldsymbol{y}_{T,s} \geq \text{OPT} \\
& \sum_{t\in R} \boldsymbol{x}_{t,s} + \boldsymbol{y}_{R,s} \leq \bar{\boldsymbol{q}}_{R,s} + \boldsymbol{\Delta}_{R,s}, \\
& \qquad s \in M, R \in Z(s) \\
& \sum_{t\in R} \boldsymbol{x}_{t,s} + \boldsymbol{y}_{R,s} \geq \underline{\boldsymbol{q}}_{R,s} + \boldsymbol{\Delta}_{R,s}, \\
& \qquad s \in M, R \in Z(s) \\
& \sum_{s\in M} \boldsymbol{x}_{t,s} + \sum_{s\in M} \boldsymbol{y}_{t,s} = C_t, \qquad t \in T \\
& \boldsymbol{x}_{t,s} \geq 0, \qquad t \in T, s \in M.
\end{aligned}
\tag{LP2}
$$

Given the above definitions, we can describe the main ideas of the algorithm. The algorithm considers the students sequentially in a given (or random) order. Each iteration consists of two steps: (a) assigning the next student to a school (the "Assignment Step") and (b) resolving fractional assignments and updating the distributional constraints (the "Resolution Step").

Assignment Step. Suppose the algorithm has assigned students $1, \ldots, i-1$. Let i be the next student to be assigned and suppose s_i is her favorite school. We need to determine whether i can be assigned to s_i. For this, we solve (LP2) with $\hat{t} = t_i$ and $\hat{s} = s_i$. If $f(t_i, s_i) \geq 1$, i is assigned to s_i (with probability 1). If $f(t_i, s_i) = 0$, i cannot be assigned to s_i. In this case the algorithm proceeds to determine whether i can be assigned to her next favorite school.

An interesting case arises when $0 < f(t_i, s_i) < 1$. In this case we must relax some of the distributional constraints to be able to assign student i to s_i. Furthermore, before observing the preferences of students who are not yet assigned, it is unclear exactly which constraints should be relaxed. Therefore at this point we assign only a fraction $f(t_i, s_i)$ of student i to s_i in the linear program and say that student i is *partially assigned*. Student i remains partially assigned until her remaining fraction is completely assigned to s_i (while the algorithm assigns other students). The assignment is considered *resolved* once she is completely assigned.

Resolution Step. After assigning a student (either partially or completely), we proceed to the next step, where the algorithm attempts to resolve any existing partial assignments and update the distributional constraints.

To explain how partial assignments are resolved, we use the following definition. A school s is *critical* for type t if $0 < f(t, s) < 1$. Consider an arbitrary student j that is partially assigned to s_j and let r_j be the remaining fraction of j that is still unassigned. We ask whether there is another school $s \neq s_j$ that is critical for type t_j. Namely, we check whether there exists a school $s \in M$ such that $0 < f(t_j, s) < 1$. If such a critical school s is found, we update the variables in a set of operations we label (j, s)-*updates*:

$$
(j,s)\text{-updates:} \qquad
\begin{cases}
\rho \leftarrow \min\big(f(t_j, s), r_j\big) \\
\Delta_{t_j,s} \leftarrow \Delta_{t_j,s} - \rho \\
\Delta_{t_j,s_j} \leftarrow \Delta_{t_j,s_j} + \rho \\
\boldsymbol{y}_{t_j,s_j} \leftarrow \boldsymbol{y}_{t_j,s_j} + \rho \\
r_j \leftarrow r_j - \rho.
\end{cases}
$$

The second-to-last operation increases the fraction of student j assigned to school s_j. Note that this ensures that j will eventually be assigned to s_j with probability 1. The second and third operations adjust the distributional constraints corresponding to schools s and s_j so they are not violated and (LP2) remains feasible. More importantly, they ensure that at every point during the algorithm, there is never more than one partially assigned student of each type.

Observe that for some critical school $s \in M$ and some subset of types $R \in Z(s)$, the lower bound quotas in (LP2) (i.e., $\underline{\boldsymbol{q}}_{R,s} + \boldsymbol{\Delta}_{R,s}$) may become negative after a set of (j, s)-updates operations. This, however, does not create a problem, because the feasible solution is required to be nonnegative. Although the upper bound quotas, $\bar{\boldsymbol{q}}_{R,s} + \boldsymbol{\Delta}_{R,s}$, can also decrease after a set of (j, s)-updates operations, they never become negative by the definition of ρ. Also observe that if for some type t, there is a less than one unit of students of type t remaining to be assigned, the outside option becomes critical for t. Therefore, while resolving a partially assigned student, we will treat the outside option as any other school.

Next we provide a formal description of the algorithm, called *serial dictatorship with dynamic menus*.

In addition to $\boldsymbol{y}$ and $\boldsymbol{\Delta}$, we also maintain a set P of partially assigned students throughout the algorithm.

Algorithm 1 (Serial Dictatorship with Dynamic Menus)

```
$\boldsymbol{\Delta} \leftarrow \vec{0}$, $\boldsymbol{y} \leftarrow \vec{0}$, $P \leftarrow \{\}$.
For $i = 1$ to $n$,
    $S \leftarrow M$.
    While $i$ is not assigned, [Assignment Step]
        $s_i \leftarrow$ $i$'s most preferred school in $S$.
        $S \leftarrow S \setminus s_i$.
        If $f(t_i, s_i) \geq 1$ then
            assign $i$ to $s_i$.
            $\boldsymbol{y}_{t_i,s_i} \leftarrow \boldsymbol{y}_{t_i,s_i} + 1$.
        Else if $0 < f(t_i, s_i) \leq 1$ then
            partially assign $i$ to $s_i$.
            $P \leftarrow P \cup \{i\}$, $r_i \leftarrow 1 - f(t_i, s_i)$.
            $\boldsymbol{y}_{t_i,s_i} \leftarrow \boldsymbol{y}_{t_i,s_i} + f(t_i, s_i)$.
    End
    While $\exists$ ($j \in P$ and $s \in M \setminus \{s_j\}$) such that
    $0 < f(t_j, s) < 1$, [Resolution Step]
        $(j, s)$-updates:
        $\rho \leftarrow \min(f(t_j, s), r_j)$.
        $\boldsymbol{\Delta}_{t_j,s} \leftarrow \boldsymbol{\Delta}_{t_j,s} - \rho$.
        $\boldsymbol{\Delta}_{t_j,s_j} \leftarrow \boldsymbol{\Delta}_{t_j,s_j} + \rho$.
        $\boldsymbol{y}_{t_j,s_j} \leftarrow \boldsymbol{y}_{t_j,s_j} + \rho$.
        $r_j \leftarrow r_j - \rho$.
        If $r_j = 0$ then
                $\triangleright$ Assignment of $j$ to $s_j$ is resolved
            $P \leftarrow P \setminus \{j\}$.
    End
End
```

Our main result is given and analyzed in Section 3.2. In the next section we illustrate Algorithm 1 on a simple example.

3.1. A Simple Example

To illustrate Algorithm 1, consider the following simple example (see the appendix for a more involved example). There are two schools s_1 and s_2. There are three students i, j, and k, whose types are t_1, t_2, and t_3, respectively. Students i and j prefer s_1 over s_2, and student k prefers s_2 over s_1. We are given the following distributional constraints. For each school $s \in \{s_1, s_2\}$ and every two types $t \neq t'$, $1 \leq \boldsymbol{x}_{t,s} + \boldsymbol{x}_{t',s} \leq 2$.

Observe that the unique feasible fractional solution is the one in which each student is assigned to each school with probability 0.5 (i.e., $\boldsymbol{x}_{t,s} = 0.5$ for every $t \in \{t_1, t_2, t_3\}$ and $s \in \{s_1, s_2\}$). Assume the order of students to be i, j and k. In the first assignment step, i is partially assigned to s_1 (with 0.5) and is added to the set of partially assigned students. In the first resolution step, s_2 is critical for type t_1 and $f(t_1, s_2) = 0.5$. Therefore, s_2 is used to resolve the assignment of i to s_1 by applying the (i, s_2)-updates. After this procedure, the constraints of schools s_1 and s_2 are updated as follows:

$$\begin{aligned} 1.5 &\leq \boldsymbol{x}_{t_1,s_1} + \boldsymbol{x}_{t_2,s_1} \leq 2.5, & 0.5 &\leq \boldsymbol{x}_{t_1,s_2} + \boldsymbol{x}_{t_2,s_2} \leq 1.5,\\ 1 &\leq \boldsymbol{x}_{t_2,s_1} + \boldsymbol{x}_{t_3,s_1} \leq 2, & 1 &\leq \boldsymbol{x}_{t_2,s_2} + \boldsymbol{x}_{t_3,s_2} \leq 2,\\ 1.5 &\leq \boldsymbol{x}_{t_3,s_1} + \boldsymbol{x}_{t_1,s_1} \leq 2.5, & 0.5 &\leq \boldsymbol{x}_{t_3,s_2} + \boldsymbol{x}_{t_1,s_2} \leq 1.5. \end{aligned}$$

In the next assignment step, because $f(t_2, s_1) = 0.5$, j will we partially assigned to s_1. Similarly, in the next resolution step, s_2 is used to resolve her assignment. The constraints are updated to:

$$\begin{aligned} 2 &\leq \boldsymbol{x}_{t_1,s_1} + \boldsymbol{x}_{t_2,s_1} \leq 3, & 0 &\leq \boldsymbol{x}_{t_1,s_2} + \boldsymbol{x}_{t_2,s_2} \leq 1,\\ 1.5 &\leq \boldsymbol{x}_{t_2,s_1} + \boldsymbol{x}_{t_3,s_1} \leq 2.5, & 0.5 &\leq \boldsymbol{x}_{t_2,s_2} + \boldsymbol{x}_{t_3,s_2} \leq 1.5,\\ 1.5 &\leq \boldsymbol{x}_{t_3,s_1} + \boldsymbol{x}_{t_1,s_1} \leq 2.5, & 0.5 &\leq \boldsymbol{x}_{t_3,s_2} + \boldsymbol{x}_{t_1,s_2} \leq 1.5. \end{aligned}$$

In the final assignment step, k will be partially assigned to s_2 because $f(t_3, s_2) = 0.5$. Because $f(t_3, s_1) = 0.5$, her assignment will be resolved in the following resolution step using the (k, s_1)-updates. The final constraints are:

$$\begin{aligned} 2 &\leq \boldsymbol{x}_{t_1,s_1} + \boldsymbol{x}_{t_2,s_1} \leq 3, & 0 &\leq \boldsymbol{x}_{t_1,s_2} + \boldsymbol{x}_{t_2,s_2} \leq 1,\\ 1 &\leq \boldsymbol{x}_{t_2,s_1} + \boldsymbol{x}_{t_3,s_1} \leq 2, & 1 &\leq \boldsymbol{x}_{t_2,s_2} + \boldsymbol{x}_{t_3,s_2} \leq 2,\\ 1 &\leq \boldsymbol{x}_{t_3,s_1} + \boldsymbol{x}_{t_1,s_1} \leq 2, & 1 &\leq \boldsymbol{x}_{t_3,s_2} + \boldsymbol{x}_{t_1,s_2} \leq 2, \end{aligned}$$

and the algorithm terminates with i and j assigned to s_1 and k assigned to s_2.

3.2. Analysis of Algorithm 1

Before we prove our main result, we establish a few useful properties of Algorithm 1.

Claim 1. *Fix a (j, s)-updates sequence of operations for student j and school s.*

i. *Let $\boldsymbol{x}$ be a feasible solution to (LP2) before the (j, s)-updates that satisfies $\boldsymbol{x}_{t_j,s} \geq \rho$. Then setting $\boldsymbol{x}_{t_j,s} \leftarrow \boldsymbol{x}_{t_j,s} - \rho$ generates a feasible solution to (LP2) after the (j, s)-updates.*

ii. *Let $\boldsymbol{x}$ be a feasible solution to (LP2) after the (j, s)-updates. Then setting $\boldsymbol{x}_{t_j,s} \leftarrow \boldsymbol{x}_{t_j,s} + \rho$ generates a feasible solution to (LP2) before the (j, s)-updates.*

Proof. We show the first part (the other part follows similar arguments). Observe that after the (j, s)-updates, $\sum_{s \in M\setminus\{\phi\}} \boldsymbol{y}_{T,s}$ increase by ρ and $\sum_{t\in T} \sum_{s\in M\setminus\{\phi\}} \boldsymbol{x}_{t,s}$ decreases by ρ, implying that the first constraint of (LP2) holds.

The second and third constraints in (LP2) hold for all schools other than s and s_j because the values of $\boldsymbol{x}$ and $\boldsymbol{y}$ and $\boldsymbol{\Delta}$ remain the same for these schools. These constraints also hold for school s because $\boldsymbol{\Delta}_{t_j,s}$ and $\boldsymbol{x}_{t_j,s}$ have decreased by the same amount, whereas the value of $\boldsymbol{y}_{t_j,s}$ remains unchanged. Similarly they hold for school s_j because $\boldsymbol{y}_{t_j,s_j}$ and $\boldsymbol{\Delta}_{t_j,s_j}$ have increased by the same amount, whereas $\boldsymbol{x}_{t_j,s_j}$ remains unchanged. The fourth constraint holds because after the (j,s)-updates, $\boldsymbol{x}_{t_j,s}$ decreases by ρ, $\boldsymbol{y}_{t_j,s_j}$ increases by ρ, and all other coordinates of $\boldsymbol{x}$ and $\boldsymbol{y}$ remain the same. Therefore, $\sum_{s\in M}\boldsymbol{x}_{t,s}+\sum_{s\in M}\boldsymbol{y}_{t,s}$ remains constant for each $t\in T$. □

Lemma 1. (LP2) *is feasible after each assignment step and after each resolution step in Algorithm* 1.

Proof. At the beginning of the algorithm, before any assignments are made, (LP2) is feasible on the basis of the assumption that there is a feasible solution for (LP1). We next show that (LP2) is feasible after each assignment step and after each (j,s)-update in the resolution step.

Consider an assignment step and let i be the student that is being assigned to school s_i (possibly partially). Observe that the only change in $\boldsymbol{y}$ when i is being assigned is $\boldsymbol{y}_{t_i,s_i}\leftarrow\boldsymbol{y}_{t_i,s_i}+\min(f(t_i,s_i),1)$. Let $\boldsymbol{x}$ be a solution for $f(t_i,s_i)$ just before assigning i to s_i. Then setting $\boldsymbol{x}_{t_i,s_i}\leftarrow\boldsymbol{x}_{t_i,s_i}-\min(\boldsymbol{x}_{t_i,s_i},1)$ while keeping all other coordinates of $\boldsymbol{x}$ the same generates a feasible solution to (LP2) immediately after i is assigned.

Now consider a resolution step and assume that $\boldsymbol{x}$ is a feasible solution of $f(t_j,s)$ just before the (j,s)-updates step, and let r be the remaining unassigned fraction of student j. Then, by Claim 1, setting $\boldsymbol{x}_{t_j,s}\leftarrow\boldsymbol{x}_{t_j,s}-\min(\boldsymbol{x}_{t_j,s},r)$ while keeping all other coordinates of $\boldsymbol{x}$ the same generates a feasible solution to (LP2) after the (j,s)-updates. □

Lemma 2. *After Algorithm* 1 *terminates, no student remains partially assigned.*

Proof. We first show that at any point during the running of algorithm, at most one student per type is partially assigned. For the sake of contradiction, suppose students i and j $(i>j)$ are the first two students of the same type that are both partially assigned at some point in the algorithm.

Consider the resolution step just before the algorithm proceeds to assign student i. By design, while the assignment of student j is not resolved and there exists a critical school s' for type t_j, the algorithm uses that school to resolve the assignment of j. Because by the end of the resolution step, the assignment of student j is not resolved, there must be no remaining critical schools for type t_j. Therefore student i cannot be partially assigned.

We can now show that no student is partially assigned after the algorithm terminates. Suppose, for the sake of contradiction, that this is not the case and there is a student j of type t that remains partially assigned, and let r_j be the fraction of j that remains. By the above argument, j is the only student of type t that is partially assigned. We claim that in the resolution step of the last iteration, assignment of j to s_j will be resolved. Suppose this is not the case; then by Lemma 1, (LP2) is still feasible after termination and has some solution $\boldsymbol{x}$. Because j is the only student of type t that is partially assigned, it must be that $\sum_{s\in M}\boldsymbol{x}_{t,s}=r_j$. Therefore, because $r_j<1$, this implies that there exists some school that is critical with respect to t, contradicting the assumption that the algorithm has terminated.

Lemma 3. *After Algorithm* 1 *terminates, the only feasible solution for* (LP2) *is* $\vec{0}$.

Proof. Assume the algorithm has terminated. By Lemma 1, (LP2) is feasible, and by Lemma 2, all students are assigned to some school (including, possibly, to the outside option) with probability 1. This implies that for every $t\in T$, $\sum_{s\in M}\boldsymbol{y}_{t,s}=C_t$. Thus $\vec{0}$ is the unique feasible solution for (LP2). □

Lemma 4. *For every* $s'\in M$ *and every* $t'\in T, f(t',s')$ *does not increase after each assignment step and each resolution step in Algorithm* 1.

Proof. Let $s'\in M$ be an arbitrary school and $t'\in T$ an arbitrary type. After an assignment step, $f(t',s')$ cannot increase because $\boldsymbol{y}$ only increases after an assignment. We next show that during the resolution step, $f(t',s')$ cannot increase after updating $\boldsymbol{\Delta}$ and $\boldsymbol{y}$ (i.e., after any (j,s)-updates). Consider a (j,s)-updates step for some student j and a critical school s for t_j. Let P_1 be the set of all feasible solutions for (LP2) right before the (j,s)-updates and P_2 be the set of all feasible solutions for (LP2) after the (j,s)-updates. Fix a feasible solution $\boldsymbol{x}\in P_2$. By Claim 1, updating $\boldsymbol{x}$ so that $\boldsymbol{x}_{t_j,s}\leftarrow\boldsymbol{x}_{t_j,s}+\rho$, where ρ is defined as in the (j,s)-updates, generates a feasible solution in P_1. Because the value of $f(t',s')$ before and after the (j,s)-updates is defined as the maximum of $\boldsymbol{x}_{t',s'}$ over all feasible solutions $\boldsymbol{x}$ in P_1 and P_2 respectively, it cannot be increasing. □

Lemma 5. *After Algorithm* 1 *terminates,* $|\boldsymbol{\Delta}_{R,s}|\le|T|$ *for every* $s\in M$ *and every* $R\in Z(s)$.

Proof. Because for any $R\in Z(s)$, $\boldsymbol{\Delta}_{R,s}=\sum_{t\in R}\boldsymbol{\Delta}_{t,s}$, it suffices to show that for any school s and for any type t, $-1\le\boldsymbol{\Delta}_{t,s}\le 1$. We show in fact that these inequalities hold at any time during the algorithm.

Let j be a student that is partially assigned to s_j. We first argue that if $\boldsymbol{\Delta}_{t_j,s_j}$ increases (line 19), it must be true that $f(t_j,s_j)=0$. To see this, note that after j is partially assigned to s_j and $\boldsymbol{y}_{t_j,s_j}$ is increased (line 13), $f(t_j,s_j)=0$, and therefore, by Lemma 4, $f(t_j,s_j)$ remains zero thereafter. Moreover, when s is critical for t_j, $\boldsymbol{\Delta}_{t_j,s}$ can only decrease. This means that for all $s'\in M$ and $t'\in T$, as long as $f(t',s')>0$, $\boldsymbol{\Delta}_{t',s'}$ can only decrease; and once $f(t',s')$ becomes zero (which happens by Lemma 3), $\boldsymbol{\Delta}_{t',s'}$ only increases.

Suppose $\Delta_{t_j,s}$ decreases by some amount ρ (at line 18) and suppose $f(t_j, s) = a$ immediately before this decrease. By Claim 1, $f(t_j, s) = a - \rho$ after the (j, s)-updates. Therefore, because school s is critical with respect to t_j, $\Delta_{t_j,s}$ can decrease by at most 1. On the other hand, Δ_{t_j,s_j} increases when we attempt to resolve the partial assignment of student j to school s_j. Let r_j be the remaining fraction of j to be assigned. Whenever Δ_{t_j,s_j} increases by ρ (line 19), r_j decreases by ρ (line 21). Therefore, by definition, Δ_{t_j,s_j} cannot increase by more than 1. □

Theorem 1. *Consider a school choice problem with distributional constraints* $\boldsymbol{q}$. *Algorithm* 1 *outputs an allocation* $\boldsymbol{y}$ *and a vector* $\boldsymbol{\Delta}$ *such that:*

i. $\boldsymbol{y}$ *is feasible with respect to* $[\underline{\boldsymbol{q}} + \boldsymbol{\Delta}, \bar{\boldsymbol{q}} + \boldsymbol{\Delta}]$,
ii. $\boldsymbol{y}$ *is allocative efficient*,
iii. $|\Delta_{R,s}| \le |T|$ *for every* $s \in M$ *and every* $R \in Z(s)$, *and*
iv. $\boldsymbol{y}$ *is Pareto efficient with respect to* $[\underline{\boldsymbol{q}} + \boldsymbol{\Delta}, \bar{\boldsymbol{q}} + \boldsymbol{\Delta}]$.

Moreover, the mechanism induced by Algorithm 1 *is strategyproof.*

Proof. The first three properties follow directly from Lemmas 1, 3, and 5.

Next we show that the mechanism induced by Algorithm 1 is strategyproof. Fix some arbitrary student i. We claim that i's preferences cannot affect the assignments of all students that are assigned before $i \in N$. Note that once a student is partially assigned to a school, she will eventually be assigned to that school with probability 1 by Lemma 2 and because the remainder of a partially assigned student j is always assigned to s_j. So the only way in which i can affect the assignment of a student $j < i$ is through her type, which cannot be altered. Finally, when it is i's turn to be assigned, she has no reason to misreport her preferences because, again, even if she is partially assigned to s_i, she will be eventually assigned to that school with probability 1.

We proceed to prove part (iv). For this we need the following two claims.

Claim 2. *Throughout the algorithm, for any* $t \in T$ *and* $s \in M$, $y_{t,s} - \Delta_{t,s}$ *does not decrease.*

Proof. Observe that during the assignment step the claim holds because $\boldsymbol{y}$ can only increase and $\boldsymbol{\Delta}$ does not change. Consider a (j, s)-updates step. Either the value of $\Delta_{t_j,s}$ decreases (line 18) while the corresponding value $y_{t_j,s}$ remains unchanged, or the values of Δ_{t_j,s_j} and y_{t_j,s_j} increase together by the same amount (lines 19–20). □

Claim 3. *Consider the values* $\boldsymbol{y}$ *and* $\boldsymbol{\Delta}$ *after a resolution step.*

i. *Suppose* $\boldsymbol{x}$ *is a feasible solution for* (LP2). *Then* $\boldsymbol{x} + \boldsymbol{y} - \boldsymbol{\Delta}$ *is an optimal feasible solution for* (LP1).

ii. *Let* $\boldsymbol{x}$ *be an optimal feasible solution for* (LP1) *such that* $\boldsymbol{x} - \boldsymbol{y} + \boldsymbol{\Delta} \ge \vec{0}$. *Then* $\boldsymbol{x} - \boldsymbol{y} + \boldsymbol{\Delta}$ *is a feasible solution for* (LP2).

Proof. We prove the first part (the second part follows from similar arguments). Note, by Lemma 1, that the set of feasible solutions for (LP2) is not empty.

Observe (from the updates in the resolution step) that $\sum_{s \in M} \Delta_{t,s} = 0$ for all $t \in T$. This implies that if $\boldsymbol{x}$ is feasible for (LP2), then $\boldsymbol{x} + \boldsymbol{y} - \boldsymbol{\Delta}$ assigns at least OPT students to regular schools because

$$\sum_{t \in T} \sum_{s \in M \setminus \{\phi\}} (\boldsymbol{x} + \boldsymbol{y} - \boldsymbol{\Delta})_{t,s} = \sum_{t \in T} \sum_{s \in M \setminus \{\phi\}} x_{t,s} + \sum_{s \in M \setminus \{\phi\}} y_{T,s} \ge \text{OPT}.$$

Because $\boldsymbol{x}$ is a feasible solution for (LP2), for every $s \in M$ and every $R \in Z(s)$,

$$\sum_{t \in R} (\boldsymbol{x} + \boldsymbol{y})_{t,s} \le \bar{q}_{R,s} + \Delta_{R,s},$$

and therefore the first constraint in (LP1) also holds, namely for every $s \in M$ and every $R \in Z(s)$,

$$\sum_{t \in R} (\boldsymbol{x} + \boldsymbol{y} - \boldsymbol{\Delta})_{t,s} \le \bar{q}_{R,s}.$$

Other constraints can be similarly verified. Moreover, by Claim 2 and because $\boldsymbol{y} \equiv \vec{0}$ and $\boldsymbol{\Delta} \equiv \vec{0}$ at the beginning of the algorithm, it must be true that $(\boldsymbol{x} + \boldsymbol{y} - \boldsymbol{\Delta})_{t,s} \ge 0$ for all $t \in T$ and $s \in M$. □

We can now complete the proof. Let $\boldsymbol{y}$ and $\boldsymbol{\Delta}$ be the outcomes of the algorithm (as in the statement of Theorem 1). For the sake of contradiction, suppose there exists an allocation $\boldsymbol{y}'$ that is feasible with respect to $[\underline{\boldsymbol{q}} + \boldsymbol{\Delta}, \bar{\boldsymbol{q}} + \boldsymbol{\Delta}]$ and Pareto dominates $\boldsymbol{y}$. Let student i be the first student (with respect to the order of the algorithm) who is assigned to different schools under $\boldsymbol{y}$ and $\boldsymbol{y}'$, and let these schools be s and s', respectively. Note that i prefers s' to s.

Let γ be the time during which the algorithm reaches the assignment step at iteration i. Let $\boldsymbol{y}_\gamma$ and $\boldsymbol{\Delta}_\gamma$ be the values of $\boldsymbol{y}$ and $\boldsymbol{\Delta}$ at time γ.

Because the algorithm did not assign i to s', it must be true that $f(t_i, s') = 0$. Therefore, by Claim 3, at time γ there is no optimal feasible solution $\boldsymbol{x}$ for (LP1) such that $\boldsymbol{x} - \boldsymbol{y}_\gamma + \boldsymbol{\Delta}_\gamma \ge \vec{0}$ and $(\boldsymbol{x} - \boldsymbol{y}_\gamma + \boldsymbol{\Delta}_\gamma)_{t_i,s'} > 0$. We will obtain a contradiction by showing that such an $\boldsymbol{x}$ exists.

We show that, upon termination, there exists $0 < c \le 1$, such that (i) $\boldsymbol{y} - \boldsymbol{\Delta} + c(\boldsymbol{y}' - \boldsymbol{y})$ is an optimal feasible solution for (LP1), (ii) $\boldsymbol{y} - \boldsymbol{\Delta} + c(\boldsymbol{y}' - \boldsymbol{y}) - \boldsymbol{y}_\gamma + \boldsymbol{\Delta}_\gamma \ge \vec{0}$, and (iii) $[\boldsymbol{y} - \boldsymbol{\Delta} + c(\boldsymbol{y}' - \boldsymbol{y}) - \boldsymbol{y}_\gamma + \boldsymbol{\Delta}_\gamma]_{t_i,s'} > 0$. This will complete the proof.

First we show that (i) holds for any $0 < c \le 1$. It is equivalent to show that there exists a c such that $0 < c \le 1$ and that $(1 - c)(\boldsymbol{y} - \boldsymbol{\Delta}) + c(\boldsymbol{y}' - \boldsymbol{\Delta})$ is an optimal feasible solution for (LP1). By Lemma 3, after termination, $\vec{0}$ is feasible for (LP2). Therefore, by Claim 3, $\boldsymbol{y} - \boldsymbol{\Delta}$ is an optimal feasible solution for (LP1). Consider (LP2) upon termination. Because $\boldsymbol{y}'$ is feasible with respect to $[\underline{\boldsymbol{q}} + \boldsymbol{\Delta}, \bar{\boldsymbol{q}} + \boldsymbol{\Delta}]$ and assigns at least OPT students (because it Pareto dominates y), $\vec{0}$ is feasible for (LP2) when y is replaced with y'. Hence, by a similar argument as in Claim 3, $y' - \Delta$ is an optimal feasible solution for (LP1). This implies (i) because for any $0 < c \leq 1$, $(1-c)(y - \Delta) + c(y' - \Delta)$ is a convex combination of optimal feasible solutions.

Next we show that (ii) holds. By Claim 2, $y - \Delta \geq y_\gamma - \Delta_\gamma$. If $y' \geq y$, then (ii) holds. Suppose this is not the case and let $s \in M$ and $t \in T$ be such that $(y' - y)_{t,s} < 0$. It is sufficient to show that for such s and t, $(y - \Delta)_{t,s} > (y_\gamma - \Delta_\gamma)_{t,s}$. Because $y' \geq y_\gamma$ and $y_{t,s} > y'_{t,s}$, we have that $y_{t,s} > (y_\gamma)_{t,s}$. Moreover, observe that when a student of type t is assigned to school s (possibly partially), $(y - \Delta)_{t,s}$ strictly increases by $\min(f(t,s), 1)$. Therefore, $(y - \Delta)_{t,s} > (y_\gamma - \Delta_\gamma)_{t,s}$, which completes the proof.

Finally, we show that (iii) holds. By Claim 2, $y - \Delta \geq y_\gamma - \Delta_\gamma$. Recall that at time γ, $f(t_i, s') = 0$. Therefore, by Lemma 4, after time γ the algorithm does not assign any other student of type t_i to s', implying that $y_{t_i,s'} = (y_\gamma)_{t_i,s'}$. Because every student $j < i$ is assigned the same school under y and y' and i is assigned to s' under y', it must be true that $y'_{t_i,s'} > y_{t_i,s'}$. This implies that (iii) holds for any $0 < c \leq 1$. □

Remarks.

1. Algorithm 1 runs in polynomial time. To see this, note that in each assignment step, we solve (LP2) at most $|M|$ times. Additionally, by Claim 1, in each (j,s)-update either the assignment of student j gets resolved or $f(t_j, s)$ becomes zero, in which case, by Lemma 4, it remains zero forever. Therefore in each resolution step, for each school $s \in M$ and each student $j \in N$, the (j,s)-updates are done at most once.

2. When Algorithm 1 selects the order in which students are assigned uniformly at random, the outcome is symmetric. That is, any two students with the same type and identical preferences have the same probabilistic assignment.

3. We have assumed that all students prefer every regular school to the outside option ϕ (and in particular all regular schools are acceptable). The results and analysis carry through, however, if the set of acceptable schools for each student is publicly known (which may be a reasonable assumption in the military or refugee assignment problems). This assumption is necessary to satisfy the lower bounds and allocative efficiency as well as to maintain strategyproofness. Indeed, if students can submit a partial preference list (e.g., by truncating their preferences), strategyproofness may fail to hold.

4. When the set of constraints are laminar and all lower- and upper-bound quotas are integers, Algorithm 1 finds an allocation that is feasible with respect to all the constraints without the need to violate any of the lower- and upper-bound quotas. We formalize this beginning with defining laminar constraints. For every $s \in M$, we say that $Z(s)$ is *laminar* if for each $R, R' \in Z(s)$, such that $R \cap R' \neq \emptyset$, either $R \subseteq R'$ or $R' \subseteq R$. We call an assignment problem laminar if $Z(s)$ is laminar for all $s \in M$.

Proposition 1. *Given a laminar assignment problem, Algorithm 1 finds an integral assignment that is feasible with respect to* q.

Before we prove the proposition, we explain how a laminar assignment problem can be reduced to a max flow problem with integer lower and upper bound capacities on edges. This will imply the existence of an integral and optimal solution of (LP1) that is feasible with respect to $\mathbf{q}$, which can also be found in polynomial time.

A few preparations are useful to illustrate this reduction. For every $s \in M \setminus \{\phi\}$ and every $R \in Z(s)$, let

$$X(R,s) = \{R' \in Z(s) | \; R' \subset R \; \text{ and } \\ (\nexists R'' \in Z(s) : R' \subset R'' \subset R)\}.$$

We now explain how to construct the graph that corresponds to the max flow problem. For every $s \in M \setminus \{\phi\}$ and every $R \in Z(s)$, add a node $u_{R,s}$. For every s and every $R \in Z(s)$ and every $R' \in X(R,s)$, add a directed edge from $u_{R',s}$ to $u_{R,s}$, with lower and upper flow constraints $\underline{q}_{R',s}$ and $\bar{q}_{R',s}$, respectively.

We add a source A and a sink B. From each node $u_{R,s}$ (not including the source or the sink), with no outgoing edges, add an edge to B with lower and upper bound flow constraints $\underline{q}_{R,s}$ and $\bar{q}_{R,s}$, respectively. We add $|T|$ *auxiliary* nodes $s_1, s_2, \ldots, s_{|T|}$, each of which represents a type and add an edge from A to each s_t with capacity C_t.

Finally, for each type t and every school $s \in M \setminus \{\phi\}$, add an edge from s_t to $u_{R,s}$ with infinite capacity for every $R \in Z(s)$ such that $t \in R$ and $u_{R,s}$ has no incoming edges.

Now, observe that every feasible solution to the laminar assignment problem (i.e., (LP1)) corresponds to a feasible flow in the constructed graph with same objective value and vice versa.

Proof of Proposition 1. Because the laminar assignment problem can be reduced to a max flow problem with integral flow constraints on the edges, the polytope corresponding to the set of optimal solutions for (LP1) has integral extreme points. We show that this also holds for (LP2) at any point during the algorithm.

First we argue that this holds if no student is ever partially assigned. To see this, note that after each student is assigned, all the lower- and upper-bound quotas change by an integral amount. Therefore, similar to the above, one can reduce our problem into a max flow problem with integral flow constraints on the edges, which implies that the polytope corresponding to the set of feasible solutions for (LP2) has integral extreme points.

It remains to show that Algorithm 1 never partially assigns any student. Suppose this is not the case and let i be the first student that is partially assigned to some school s_i. Just before assigning i, let $\mathcal{P}$ be the set of feasible solutions for (LP2). Observe that by the above argument, all extreme points of $\mathcal{P}$ are integral. Because i was partially assigned to s_i, this implies that

$$0 < \max_{x \in \mathcal{P}} x_{t_i,s} < 1.$$

This is a contradiction, because every point in $\mathcal{P}$ can be written as a convex combination of extreme points.

This means that there must exist an integral point $x' \in \mathcal{P}$ with $x'_{t_i,s} = 1$. □

4. Generalized Probabilistic Serial Mechanism

In this section we generalize the probabilistic serial mechanism (PS) to allow for distributional constraints. The PS mechanism was introduced by Bogomolnaia and Moulin (2001), who showed that it satisfies several desirable properties such as ordinal efficiency, envy-freeness, and weak strategyproofness.

Let us begin with a brief description of PS, also known as the *eating* algorithm. Treating each school (including the outside option) as a divisible object, the algorithm asks the students to eat from schools simultaneously and at the same rate until each consumes one unit. Every student begins eating from her favorite school. Whenever a school s is fully consumed, students who were eating from s then proceed to eat from their next preferred available school. The process concludes when each student consumes one unit. The resulting fractional outcome is implemented using a lottery over allocations defined by the Birkhoff-Von-Neumann Theorem (Schrijver 2003).

To define the properties of PS, we use the (student) assignment variables $x_{i,s}$, which is interpreted as the probability that student i is assigned to school s. For assignment x, we denote by $x_i = (x_{i,s})_{s\in S}$ the assignment for student i, which is the distributional outcome for student i.

Let x and y be assignments. We say that x_i *stochastically dominates* y_i with respect to preference order $>_i$, if for every $s \in M$:

$$\sum_{s':s'>_i s} x_{i,s'} \geq \sum_{s':s'>_i s} y_{i,s'},$$

in which case we write $x_i sd(>_i) y_i$.

Given a preference profile $(>_i)_{i\in N}$, we say that y is *stochastically dominated* by x if $x_i sd(>_i) y_i$ for all $i \in N$ and $x \neq y$. The assignment x is said to be *ordinally efficient*, if it is a feasible solution (for (LP3)) and it is not stochastically dominated by any other feasible assignment.

The assignment x is *within-type envy-free* if for any two students i, j of the same type t and every school $s \in M$:

$$\sum_{s':s'>_i s} x_{i,s'} \geq \sum_{s':s'>_i s} x_{j,s'}.$$

Finally, a mechanism is *weakly strategyproof* if for every student i and any preference profile of all other students, reporting $>_i$ results in an assignment x_i for i, that is not stochastically dominated with respect to $>_i$ by any other assignment x'_i for i that can be obtained by i misreporting her preferences, unless $x_i = x'_i$.

Our algorithm, which we call the *generalized probabilistic serial* (GPS), generalizes the PS algorithm to the setting with distributional constraints. The outcome of the GPS is a fractional assignment that does not violate any lower- or upper-bound quotas. We show that such a fractional assignment can be implemented as a lottery over integral solutions that violate each quota by at most $|T|$.

We begin with establishing the implementation of such a fractional solution as a distribution over allocations and explain the details of the GPS algorithm in Section 4.2.

4.1. Implementing a Fractional Solution as a Lottery over Allocations

Recall the linear program (LP1) for optimizing allocative efficiency given the distributional constraints. This program can be rewritten using the (student) assignment variables $x_{i,s}$, which can be interpreted as the probability that student i is assigned to school s:

$$\begin{aligned}
\mathrm{OPT} = \max_{x\in\mathbb{R}^{N\times M}} \quad & \sum_{i\in N}\sum_{s\in M\setminus\{\phi\}} x_{i,s} \\
\text{s.t.} \quad & \sum_{i\in N: t_i \in R} x_{i,s} \leq \bar{q}_{R,s}, && \forall s \in M, R \in Z(s) \\
& \sum_{i\in N: t_i \in R} x_{i,s} \geq \underline{q}_{R,s}, && \forall s \in M, R \in Z(s) \\
& \sum_{s\in M} x_{i,s} = 1, && \forall i \in N \\
& x_{i,s} \geq 0, && \forall i \in N, s \in M.
\end{aligned} \tag{LP3}$$

We will show that every optimal solution for (LP3) can be written as a convex combination of approximately feasible allocations.

Definition 1. An allocation x is *approximately feasible* if it assigns students in a way that

1. Each lower- and upper-bound quota is violated by at most $|T|$. That is,

$$\underline{q}_{R,s} - |T| \leq \sum_{t\in R} x_{t,s} \leq \bar{q}_{R,s} + |T| \quad \forall s \in M, R \in Z(s).$$

2. At least $\lfloor \mathrm{OPT} \rfloor$ students are assigned to regular schools.

Lemma 6. *Every optimal solution for (LP3) can be written as a convex combination of approximately feasible integral solutions.*

Proof. Let $\boldsymbol{x}$ be an optimal solution for (LP3). Consider the following polyhedron with variables $\boldsymbol{y}$:

$$\left\lfloor \sum_{i\in N: t_i=t} x_{i,s} \right\rfloor \le \sum_{i\in N: t_i=t} y_{i,s} \le \left\lceil \sum_{i\in N: t_i=t} x_{i,s} \right\rceil, \qquad \forall t\in T, s\in M\setminus\{\phi\}$$

$$\left\lfloor \sum_{i\in N} x_{i,\phi} \right\rfloor \le \sum_{i\in N} y_{i,\phi} \le \left\lceil \sum_{i\in N} x_{i,\phi} \right\rceil$$

$$\sum_{s\in M} y_{i,s} = 1, \qquad \forall i\in N$$

$$y_{i,s}\ge 0, \qquad \forall i\in N, s\in M.$$

Note that $\boldsymbol{x}$ is a feasible solution for the above linear program, so the set of feasible solutions is nonempty. Second, the above linear program corresponds to a generalized assignment problem with integer lower- and upper-bound capacities, and therefore the corner points of its corresponding polytope are integral.

Finally, every corner point $\boldsymbol{y}$ of the above polytope is approximately feasible. The reason is that for each upper-bound quota $\bar{q}_{R,s}$ over $s\in M\setminus\{\phi\}$ and $R\in Z(s)$, we have

$$\sum_{i\in N: t_i\in R} y_{i,s} \le \sum_{t\in R}\left\lceil \sum_{i\in N: t_i=t} x_{i,s}\right\rceil \le \sum_{t\in R}\left(\sum_{i\in N: t_i=t} x_{i,s}+1\right) \le \sum_{i\in N: t_i\in R} x_{i,s}+|T| \le \bar{q}_{R,s}+|T|.$$

A similar argument can be used for lower-bound quotas. Furthermore, the number of students who are assigned to the outside option by $\boldsymbol{y}$ is:

$$\sum_{i\in N} y_{i,\phi} \le \left\lceil \sum_{i\in N} x_{i,\phi}\right\rceil.$$

Therefore at least $\lfloor OPT \rfloor$ students are assigned to regular schools. □

We note that Lemma 6 could also be derived from (Akbarpour and Nikzad 2019). Additionally, as remarked in the previous section, if the set of constraints are bihierarchical, one can implement the assignment without violating any lower- or upper-bound quota.

4.2. A Generalization of the Probabilistic Serial Mechanism

The eating algorithm, PS, is very similar to Bogomolnaia and Moulin (2001) in that every student eats from her favorite school as long the "partial" assignment is "extendable" to an optimal solution for (LP3) and switches to her next favorite school when a constraint becomes tight.

Definition 2. A vector $\boldsymbol{y}$ is *extendable* if there exists $\boldsymbol{x}$, a feasible solution for (LP3) that also satisfies the following conditions:

i. $\boldsymbol{x}$ dominates $\boldsymbol{y}$ (i.e., $x_{i,s}\ge y_{i,s}$ for all $i\in N$ and $s\in M\setminus\{\phi\}$), and

ii. $\boldsymbol{x}$ is allocative efficient (i.e., $\sum_{i\in N}\sum_{s\in M\setminus\{\phi\}} x_{i,s} =$ OPT).

Given any vector $\boldsymbol{y}$, it is possible to check whether it is extendable by adding the linear constraints corresponding to conditions (i) and (ii) to (LP3) and testing whether the set of feasible solutions remains nonempty. Let $\mathscr{E}$ denote the set of extendable vectors.

Similar to PS, the GPS starts from an empty assignment and asks every student to eat from the schools at a constant rate in their order of preference. The main difference, however, is that we keep the assignment vector in $\mathscr{E}$ at all times. Whenever we reach a boundary at which we are about to leave the set $\mathscr{E}$, we prevent the corresponding students from consuming their current school and ask them to move to their next most preferred available school. We then continue the process and have all agents consume their current preferred and available school. We do this until the algorithm terminates (i.e., every student has consumed one unit).

The above process can be implemented as follows. Consider any point during the running of the algorithm and let $x_{i,s}$ denote how much of school s is consumed by student i at that time. Also consider $\boldsymbol{\theta}$, which denotes the eating pattern of the students at this time (i.e., $\boldsymbol{\theta}_{i,s}=1$ if i is currently eating from s and 0 otherwise). Using a linear program, we can find the maximum c such that $\boldsymbol{x}+c\boldsymbol{\theta}$ is extendable. This will determine how long we can continue the current eating pattern. Whenever we reach that point, a new constraint will be tight. At that point, we ask all the students involved in that constraint to stop eating their current school and move to their next option. We repeat this procedure until all students have consumed one unit in which point $\boldsymbol{x}$ is a (fractional) optimal solution for (LP3).

The running time of this process is polynomial, because every student switches at most $|M|$ times. Once a student reaches the outside option, she can continue consuming that option until her total consumption reaches one unit.

Theorem 2. *The (fractional) assignment produced by the generalized probabilistic serial (GPS) algorithm is*

i. *within-type envy-free,*

ii. *ordinally efficient, and*

iii. *implementable using a lottery over approximately feasible assignments.*

Proof. The following claim will be needed:

Claim 4. *Once a student of type t is prevented from eating from some school s, then no students of type t will ever be able to eat from that school again.*

Proof. Consider the first time the algorithm starts blocking students of type t from eating school s. Let $\boldsymbol{y}$ denote the set of probability shares consumed by students up to that point such that $\boldsymbol{y}_{i,s}$ represents how much of school s is consumed by student i. Because the algorithm treats all students of the same type the same way, at that point all the students of type t are blocked from consuming s. By Definition 2, this means that there exists no vector $\boldsymbol{y}'$ such that $\boldsymbol{y}' \geq \boldsymbol{y}$ and $\boldsymbol{y}'_{i,s} > \boldsymbol{y}_{i,s}$ for some student i with type t. Now, because after this point the vector $\boldsymbol{y}$ can only increase, we can never have a vector $\boldsymbol{y}'$ that satisfies this property, and so students of type t will always remain blocked. □

To show within-type envy-freeness, we use proof by contradiction. Assume there exists a student i that envies another student j of the same type. This implies that there exists some school s such that:

$$\sum_{s':s' \succ_i s} \boldsymbol{x}_{j,s'} > \sum_{s':s' \succ_i s} \boldsymbol{x}_{i,s'}.$$

Let $A = \{s' \in M | s' \succ_i s\}$. We know that while it is possible for student i to eat schools in A, she does not consume any schools in $M \setminus A$. Additionally, by Claim 4, we know that once she is blocked from eating some school in A, no other student of type t can ever consume from that school. Because all the students consume schools at the same rate, this implies that $\sum_{s':s' \succ_i s} \boldsymbol{x}_{j,s'} \leq \sum_{s':s' \succ_i s} \boldsymbol{x}_{i,s'}$ for all $j \in N$ with type t_i, which is a contradiction.

To show ordinal efficiency, we use proof by contradiction. Let $\boldsymbol{x}$ be the fractional assignment obtained by the eating algorithm. Consider another feasible assignment $\boldsymbol{y}$ that stochastically dominates $\boldsymbol{x}$. Note that the values of all the entries of $\boldsymbol{x}$ have evolved over time and were initialized all to zero. Let $\boldsymbol{x}_0$ denote the value of $\boldsymbol{x}$ the last time $\boldsymbol{x}$ was such that $\boldsymbol{x} \leq \boldsymbol{y}$. At that time, there must exist a student i who is eating a school s and $(\boldsymbol{x}_0)_{i,s} = \boldsymbol{y}_{i,s}$. At that point, student i had the choice to eat any school s' such that $(\boldsymbol{x}_0)_{i,s'} < \boldsymbol{y}_{i,s'}$, but she preferred to keep eating school s. Because $\boldsymbol{x}_{i,s} > \boldsymbol{y}_{i,s}$, $\boldsymbol{y}_i$ cannot stochastically dominate $\boldsymbol{x}_i$ with respect to $\succ_i$, which is a contradiction.

Property (iii) is a direct implication of Lemma 6. □

We note that Katta and Sethuraman (2006) generalize PS to weak preferences without distributional constraints. Their mechanism resembles ours in the sense that it updates "menus" by resolving flow problems.

To illustrate GPS, consider the example in Section 3.1. Each student first eats from her favorite school (at rate 1). After half a unit of time, each student has consumed half of her favorite school, and the vector of consumed probability shares is no longer extendable. All students then switch to eat from their next favorite school. Again, after half a unit of time, each student has consumed half of their second favorite school, and now each student has consumed one unit. Finally, the fractional assignment is implemented as a lottery over approximately feasible allocations.

4.3. An Impossibility Result

Bogomolnaia and Moulin (2001) find that no mechanism is ordinally efficient, envy-free, and strategyproof in the context of allocating indivisible objects to homogeneous agents. However, they show that the PS mechanism is weakly strategyproof. We prove that with distributional constraints, weak strategyproofness cannot hold when both within-type envy-freeness and ordinal efficiency are required. [A similar impossibility for the allocation problem without distributional constraints was established for the case in which preferences are not necessarily strict (Katta and Sethuraman 2006).]

Theorem 3. *In the school choice problem with distributional constraints, no mechanism is ordinally efficient, within-type envy-free, and weakly strategyproof.*

Proof. Consider three schools s_1, s_2, s_3 and two students i, j of type t. There are three additional types t_1, t_2, t_3 other than t and one student of each of those types. We have the following constraints for $s \in \{s_1, s_2\}$:

$$\begin{aligned} 1 \leq \boldsymbol{x}_{t_1,s} + \boldsymbol{x}_{t_2,s} \leq 1, \\ 1 \leq \boldsymbol{x}_{t_2,s} + \boldsymbol{x}_{t_3,s} \leq 1, \\ 1 \leq \boldsymbol{x}_{t_3,s} + \boldsymbol{x}_{t_1,s} \leq 1, \\ 0 \leq \boldsymbol{x}_{t,s} + \boldsymbol{x}_{t_1,s} \leq 1, \end{aligned}$$

where $\boldsymbol{x}_{t,s}$ represents the number of students of type t assigned to school s. Note that in the above inequalities, the values of $\boldsymbol{x}_{t_1,s}, \boldsymbol{x}_{t_2,s}$, and $\boldsymbol{x}_{t_3,s}$ are all uniquely determined to be 0.5. Therefore, in each school s_1 and s_2, there is an upper bound of 0.5 for type t. School s_3 imposes no constraints.

Now suppose that i and j have the following preferences:

$$\succ_i: s_1 \succ_i s_2 \succ_i s_3,$$
$$\succ_j: s_2 \succ_j s_3 \succ_j s_1.$$

Assume j reports her true preferences and i misreports $\succ'_i: s_2 \succ'_i s_1 \succ'_i s_3$. By ordinal efficiency, both s_1 and s_2 should be filled up to 0.5 with students i and j. In addition, s_1 should be filled with student i. Because of within-type envy-freeness, i and j must each be assigned 0.25 of s_2. Therefore i's assignment in schools s_1, s_2, s_3 must be $(0.5, 0.25, 0.25)$ and j's assignment must be $(0, 0.25, 0.75)$. So when i and j both report truthfully, i must be assigned 0.5 of s_1 and at least

0.25 of s_2. This implies that j is assigned at most 0.25 of s_2. Denote the assignment of j in this case by x_j.

Using this observation, we claim that j will benefit from misreporting her preferences. Assume i reports her true preferences and j misreports $\succ'_j$: $s_2 \succ'_j s_1 \succ'_j s_3$. Denote the assignment under these reports by x'. By ordinal efficiency, s_1 and s_2 should be filled up to 0.5. Moreover, i and j cannot both have positive assignment probabilities to s_1 and s_2 because they will benefit from exchanging probability shares. In combination with within-type envy-freeness, this implies that j's assignment must be $(0, 0.5, 0.5)$. This means that $x'_j \neq x_j$ and $x'_j (sd)_{\succ_j} x_j$, contradicting weak strategyproofness. □

5. Conclusion

We studied the assignment problem under distributional constraints and privately known preferences. There may be numerous Pareto-efficient assignments, which can vary significantly in the number of assigned students. The mechanisms we introduced result in assignments that may violate each lower- and upper-bound quota by at most the number of students' types but can assign as many students as can be assigned via a fractional solution. Although our generalization of the serial dictatorship is strategyproof, we demonstrate that distributional constraints introduce a new barrier to achieving weak strategyproofness in combination with within-type envy-freeness and ordinal efficiency.

Appendix. Illustration of the Serial Dictatorship Algorithm with Dynamic Menus

In Section 3 we illustrated the performance of Algorithm 1 on a simple example. Here, we expand on that by analyzing a more detailed example. Suppose there are two schools s_1 and s_2 and seven students $i_1, i_2, \ldots, i_7$ whose types are t_1, t_2, t_3, t_4, t_5, t_1, and t_2, respectively. Students i_2 and i_4 prefer s_2 over s_1, and the rest of the students prefer s_1 over s_2. We are given the following distributional constraints:

$$\begin{array}{ll}
1 \le x_{t_1,s_1} + x_{t_2,s_1} \le 1, & 1 \le x_{t_3,s_2} + x_{t_4,s_2} \le 1,\\
1 \le x_{t_2,s_1} + x_{t_3,s_1} \le 1, & 1 \le x_{t_4,s_2} + x_{t_5,s_2} \le 1,\\
1 \le x_{t_3,s_1} + x_{t_1,s_1} \le 1, & 1 \le x_{t_5,s_2} + x_{t_3,s_2} \le 1,\\
0 \le x_{t_1,s_1} + x_{t_2,s_1} + x_{t_3,s_1} \le 2, & 0 \le x_{t_1,s_2} + x_{t_2,s_2} + x_{t_3,s_2} \le 2.
\end{array}$$

Observe that given these quotas, in any feasible solution, each of types t_1, t_2, and t_3 will be assigned to school s_1 with probability 0.5, and each of types t_3, t_4, and t_5 will be assigned to school s_2 with probability 0.5 as well. Assume the order of students to be $i_1, i_2, \ldots, i_7$. In the first assignment step, i_1 is partially assigned to s_1 (with 0.5) and is added to the set of partially assigned students. In the first resolution step, no school is critical for type t_1.

In the second round, i_2 is completely assigned to s_2. This time, in the resolution step, s_2 is critical for type t_1 and $f(t_1, s_2) = 0.5$. Therefore, s_2 is used to resolve the assignment of i_1 to s_1 by applying the (i_1, s_2)-updates. After this procedure, the constraints of schools s_1 and s_2 are updated as follows:

$$\begin{array}{ll}
1.5 \le x_{t_1,s_1} + x_{t_2,s_1} \le 1.5, & 1 \le x_{t_3,s_2} + x_{t_4,s_2} \le 1,\\
1 \le x_{t_2,s_1} + x_{t_3,s_1} \le 1, & 1 \le x_{t_4,s_2} + x_{t_5,s_2} \le 1,\\
1.5 \le x_{t_3,s_1} + x_{t_1,s_1} \le 1.5, & 1 \le x_{t_5,s_2} + x_{t_3,s_2} \le 1,\\
0.5 \le x_{t_1,s_1} + x_{t_2,s_1} + x_{t_3,s_1} \le 2.5, & \\
& -0.5 \le x_{t_1,s_2} + x_{t_2,s_2} + x_{t_3,s_2} \le 1.5.
\end{array}$$

In the next assignment step, because $f(t_3, s_1) = 0.5$, i_3 is partially assigned to s_1. In the following resolution step, s_2 is used to resolve her assignment and after the (i_3, s_2)-updates the constraints are updated to:

$$\begin{array}{ll}
1.5 \le x_{t_1,s_1} + x_{t_2,s_1} \le 1.5, & 0.5 \le x_{t_3,s_2} + x_{t_4,s_2} \le 0.5,\\
1.5 \le x_{t_2,s_1} + x_{t_3,s_1} \le 1.5, & 1 \le x_{t_4,s_2} + x_{t_5,s_2} \le 1,\\
2 \le x_{t_3,s_1} + x_{t_1,s_1} \le 2, & 0.5 \le x_{t_5,s_2} + x_{t_3,s_2} \le 0.5,\\
1 \le x_{t_1,s_1} + x_{t_2,s_1} + x_{t_3,s_1} \le 3, & \\
& -1 \le x_{t_1,s_2} + x_{t_2,s_2} + x_{t_3,s_2} \le 1.
\end{array}$$

Next, we have $f(t_4, s_2) = 0.5$, and therefore i_4 is partially assigned to s_2. In the resolution step s_1 is used to resolve this partial assignment. After the (i_4, s_1)-updates the constraints are:

$$\begin{array}{ll}
1.5 \le x_{t_1,s_1} + x_{t_2,s_1} \le 1.5, & 1 \le x_{t_3,s_2} + x_{t_4,s_2} \le 1,\\
1.5 \le x_{t_2,s_1} + x_{t_3,s_1} \le 1.5, & 1.5 \le x_{t_4,s_2} + x_{t_5,s_2} \le 1.5,\\
2 \le x_{t_3,s_1} + x_{t_1,s_1} \le 2, & 0.5 \le x_{t_5,s_2} + x_{t_3,s_2} \le 0.5,\\
1 \le x_{t_1,s_1} + x_{t_2,s_1} + x_{t_3,s_1} \le 3, & \\
& -1 \le x_{t_1,s_2} + x_{t_2,s_2} + x_{t_3,s_2} \le 1.
\end{array}$$

In the next assignment step, we have $f(t_5, s_1) = 0.5$ and i_5 is partially assigned to s_1. In the resolution step, s_2 is used to resolve this assignment. After the (i_5, s_2) updates, the constraints are:

$$\begin{array}{ll}
1.5 \le x_{t_1,s_1} + x_{t_2,s_1} \le 1.5, & 1 \le x_{t_3,s_2} + x_{t_4,s_2} \le 1,\\
1.5 \le x_{t_2,s_1} + x_{t_3,s_1} \le 1.5, & 1 \le x_{t_4,s_2} + x_{t_5,s_2} \le 1,\\
2 \le x_{t_3,s_1} + x_{t_1,s_1} \le 2, & 0 \le x_{t_5,s_2} + x_{t_3,s_2} \le 0,\\
1 \le x_{t_1,s_1} + x_{t_2,s_1} + x_{t_3,s_1} \le 3, & \\
& -1 \le x_{t_1,s_2} + x_{t_2,s_2} + x_{t_3,s_2} \le 1.
\end{array}$$

Next, we have $f(t_1, s_1) = 0$ and $f(t_1, s_2) = 0$ and therefore i_6 is assigned to the outside option. Finally, when i_7 is about to get assigned, we have $f(t_2, s_1) = 0.5$ and therefore i_7 is partially assigned to s_1. In the following resolution step, we use the outside option, that is ϕ to resolve this partial assignment. After the (i_7, ϕ)-updates, the constraints are:

$$\begin{array}{ll}
2 \le x_{t_1,s_1} + x_{t_2,s_1} \le 2, & 1 \le x_{t_3,s_2} + x_{t_4,s_2} \le 1,\\
2 \le x_{t_2,s_1} + x_{t_3,s_1} \le 2, & 1 \le x_{t_4,s_2} + x_{t_5,s_2} \le 1,\\
2 \le x_{t_3,s_1} + x_{t_1,s_1} \le 2, & 0 \le x_{t_5,s_2} + x_{t_3,s_2} \le 0,\\
1.5 \le x_{t_1,s_1} + x_{t_2,s_1} + x_{t_3,s_1} \le 3.5, & \\
& -1 \le x_{t_1,s_2} + x_{t_2,s_2} + x_{t_3,s_2} \le 1.
\end{array}$$

References

Abdulkadiroğlu A, Pathak PA, Roth AE (2009) Strategy-proofness vs. efficiency in matching with indifferences: Redesigning the NYC high school match. *Amer. Econom. Rev.* 99(5):1954–1978.

Akbarpour M, Nikzad A (2019) Approximate random allocation mechanisms. Working paper, Stanford University, Stanford, CA.

Ashlagi I, Saberi A, Shameli A (2019) Assignment mechanisms under distributional constraints. *Proc. 30th Annual ACM-SIAM Sympos. Discrete Algorithms* (Society for Industrial and Applied Mathematics, Philadelphia), 229–240.

Baswana S, Chakrabarti P, Chandran S, Kanoria Y, Patange U (2019) Centralized admissions for engineering colleges in India. *INFORMS J. Appl. Analytics* 49(5):338–354.

Biró P, Fleiner T, Irving RW, Manlove DF (2010) The college admissions problem with lower and common quotas. *Theoret. Comput. Sci.* 411(34):3136–3153.

Bogomolnaia A, Moulin H (2001) A new solution to the random assignment problem. *J. Econom. Theory* 100(2):295–328.

Braun S, Dwenger N, Kübler D, Westkamp A (2014) Implementing quotas in university admissions: An experimental analysis. *Games Econom. Behav.* 85:232–251.

Budish E, Che YK, Kojima F, Milgrom P (2013) Designing random allocation mechanisms: Theory and applications. *Amer. Econom. Rev.* 103(2):585–623.

Chen Y, Kesten O (2017) Chinese college admissions and school choice reforms: A theoretical analysis. *J. Political Econom.* 125(1): 99–139.

Chen Y, Snmez T (2002) Improving efficiency of on-campus housing: An experimental study. *Amer. Econom. Rev.* 92(5):1669–1686.

Delacrétaz D, Kominers SD, Teytelboym A (2016) Refugee resettlement. Working paper, University of Oxford, Oxford, UK.

Echenique F, Yenmez MB (2015) How to control controlled school choice? *Amer. Econom. Rev.* 105(8):2679–2694.

Ehlers L, Hafalir IE, Yenmez MB, Yildirim MA (2014) School choice with controlled choice constraints: Hard bounds vs. soft bounds. *J. Econom. Theory* 153:648–683.

Fleiner T, Kamiyama N (2012) A matroid approach to stable matchings with lower quotas. *Proc. 23rd Annual ACM-SIAM Sympos. Discrete Algorithms* (Society for Industrial and Applied Mathematics, Philadelphia), 135–142.

Fragiadakis D, Iwasaki A, Troyan P, Ueda S, Yokoo M (2016) Strategyproof matching with minimum quotas. *ACM Trans. Econom. Comput.* 4(1):6:1–6:40.

Fragiadakis D, Troyan P (2017) Improving matching under hard distributional constraints. *Theoret. Econom.* 12(2):863–908.

Goto M, Kurata R, Hamada N, Iwasaki A, Yokoo M (2015) Improving fairness in nonwasteful matching with hierarchical regional minimum quotas. *Proc. 2015 Internat. Conf. Autonomous Agents and Multiagent Systems* (International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC), 1887–1888.

Hafalir IE, Yenmez MB, Yildirim MA (2013) Effective affirmative action in school choice. *Theoret. Econom.* 8(2):325–363.

Hamada K, Iwama K, Miyazaki S (2016) The hospitals/residents problem with lower quotas. *Algorithmica* 74(1):440–465.

Huang CC (2010) Classified stable matching. *Proc. 21st Annual ACM-SIAM Sympos. Discrete Algorithms* (Society for Industrial and Applied Mathematics, Philadelphia), 1235–1253.

Kamada Y, Kojima F (2014) Efficient matching under distributional constraints: Theory and applications. *Amer. Econom. Rev.* 105(1): 67–99.

Kamada Y, Kojima F (2017) Stability concepts in matching under distributional constraints. *J. Econom. Theory* 168:107–142.

Katta AK, Sethuraman J (2006) A solution to the random assignment problem on the full preference domain. *J. Econom. Theory* 131(1): 231–250.

Kojima F (2012) School choice: Impossibilities for affirmative action. *Games Econom. Behav.* 75(2):685–693.

Kominers SD, Sönmez T (2013) Designing for diversity in matching. *Proc. 14th ACM Conf. Econom. Comput.* (ACM, New York), 603–604.

Nguyen T, Vohra R (2017) Stable matching with proportionality constraints. *Proc. 2017 ACM Conf. Econom. Comput*. (Association for Computing Machinery, New York), 675–676.

Nguyen T, Vohra R (2019) Near feasible stable matchings with complementarities. *Amer. Econom. Rev.* 108(11):3154–3169.

Noda S (2018) Large matchings in large markets with flexible supply. Working paper, University of British Columbia, Vancouver.

Pathak PA, Sönmez T (2013) School admissions reform in Chicago and England: Comparing mechanisms by their vulnerability to manipulation. *Amer. Econom. Rev.* 103(1):80–106.

Schrijver A (2003) *Combinatorial Optimization: Polyhedra and Efficiency*, vol. 24 (Springer Science & Business Media, New York).

Sönmez T, Switzer TB (2013) Matching with (branch-of-choice) contracts at the United States military academy. *Econometrica* 81(2):451–488.

Westkamp A (2013) An analysis of the German university admissions system. *Econom. Theory* 53(3):561–589.

Yokoi Y (2016) A generalized polymatroid approach to stable matchings with lower quotas. *Math. Oper. Res.* 42(1):238–255.

Itai Ashlagi is an associate professor at the department of management science and engineering at Stanford University. He is interested in game theory and market design. He is the recipient of the NSF CAREER award and a Franz Edelman Laureate.

Amin Saberi is associate professor in the department of management science and engineering at Stanford University. His research centers on algorithm design, especially approximation algorithms and applications in games, networks, and markets. He is a recipient of the Terman Fellowship, Alfred Sloan Fellowship and a number of best paper awards.

Ali Shameli is a PhD student in the MS&E Department at Stanford University. Prior to Stanford, he completed his bachelor's degree in computer engineering at Sharif University of Technology. His research is mostly in design and analysis of algorithms, operations research and optimization. He is the recipient of the best paper award for WINE 2017.