This article was downloaded by: [160.39.9.63] On: 01 March 2022, At: 09:34

Publisher: Institute for Operations Research and the Management Sciences (INFORMS)

INFORMS is located in Maryland, USA



# **Operations Research**

Publication details, including instructions for authors and subscription information: <a href="http://pubsonline.informs.org">http://pubsonline.informs.org</a>

# Legal Assignments and Fast EADAM with Consent via Classic Theory of Stable Matchings

Yuri Faenza, Xuan Zhang

#### To cite this article:

Yuri Faenza, Xuan Zhang (2022) Legal Assignments and Fast EADAM with Consent via Classic Theory of Stable Matchings. Operations Research

Published online in Articles in Advance 24 Jan 2022

. <a href="https://doi.org/10.1287/opre.2021.2199">https://doi.org/10.1287/opre.2021.2199</a>

Full terms and conditions of use: <a href="https://pubsonline.informs.org/Publications/Librarians-Portal/PubsOnLine-Terms-and-Conditions">https://pubsonline.informs.org/Publications/Librarians-Portal/PubsOnLine-Terms-and-Conditions</a>

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact permissions@informs.org.

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2022, INFORMS

Please scroll down for article—it is on subsequent pages



With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes.

For more information on INFORMS, its publications, membership, or meetings visit <a href="http://www.informs.org">http://www.informs.org</a>

Articles in Advance, pp. 1–18
ISSN 0030-364X (print), ISSN 1526-5463 (online)

#### **Methods**

# Legal Assignments and Fast EADAM with Consent via Classic Theory of Stable Matchings

#### Yuri Faenza, Xuan Zhanga

<sup>a</sup> Department of Industrial Engineering and Operations Research, Columbia University, New York, New York 10027 Contact: yf2414@columbia.edu, b https://orcid.org/0000-0002-3148-2159 (YF); xz2569@columbia.edu, https://orcid.org/0000-0002-6535-5449 (XZ)

Received: October 24, 2018

Revised: February 24, 2020; February 3, 2021

Accepted: August 11, 2021

Published Online in Articles in Advance:

January 24, 2022

Area of Review: Optimization

https://doi.org/10.1287/opre.2021.2199

Copyright: © 2022 INFORMS

**Abstract.** Gale and Shapley's *stable assignment problem* has been extensively studied, applied, and extended. In the context of school choice, mechanisms often aim at finding an assignment that is more favorable to students. We investigate two extensions introduced in this framework—legal assignments and the efficiency adjusted deferred acceptance mechanism (EADAM) algorithm—through the lens of the classic theory of stable matchings. In any instance, the set  $\mathcal L$  of legal assignments is known to contain all stable assignments. We prove that  $\mathcal L$  is exactly the set of stable assignments in another instance. Moreover, we show that essentially all optimization problems over  $\mathcal L$  can be solved within the same time bound needed for solving it over the set of stable assignments. A key tool for this latter result is an algorithm that finds the student-optimal legal assignment. We then generalize our algorithm to obtain the assignment output of EADAM with any given set of consenting students without sacrificing the running time, hence largely improving in both theory and practice over known algorithms. Finally, we show that the set  $\mathcal L$  can be much larger than the set of stable matchings, connecting legal matchings with certain concepts and open problems in the literature.

Funding: Y. Faenza is supported by the National Science Foundation award 2046146 and X. Zhang is partially funded by a fellowship from the Cheung Kong Graduate School of Business.
Supplemental Material: The e-companion is available at https://doi.org/10.1287/opre.2021.2199.

Keywords: stable matchings • distributive lattice • rotations • school choice problem • legal assignments • EADAM algorithm • Latin marriages

### 1. Introduction

Stable matchings and stable assignments are fundamental paradigms in operations research and the design of matching markets. Since the seminal work of Gale and Shapley (1962), stable assignments have received widespread attention for their mathematical elegance and broad applicability (see, e.g., Gusfield and Irving 1989, Roth and Sotomayor 1992, Manlove 2013). Those two facets are tightly connected. For instance, a detailed understanding of the lattice structure of stable matchings led to many fast algorithms for, for example, enumerating all stable matchings (Gusfield 1987) and finding a stable matching that maximizes some linear profit function (Irving et al. 1987). In turn, these algorithmic results propelled the application of stable matchings to many markets, such as college admission, assigning residents to hospitals (Roth 1984), and kidney transplant (Roth et al. 2005).

One of the most important applications of matching theory, the school choice problem, considers the assignment of high school students to public schools. After the pioneering work of Abdulkadiroğlu and Sönmez (2003), many school districts, such as New York City and Boston, adopted the student-optimal stable mechanism for its fairness (no priority violation or stability) and *strategy-proofness* (for students). The mechanism asks students to report their (strict) preferences of the schools and schools to report their priorities<sup>1</sup> (preferences with ties) over the students. It then randomly breaks ties in the latter to obtain an instance of the stable assignment problem and performs Gale–Shapley's algorithm<sup>2</sup> to obtain the studentoptimal stable assignment. Gale-Shapley's algorithm embodies many desirable qualities an algorithm can have: it is simple and elegant, runs in time linear in the size of the instance, and outputs an assignment that satisfies the aforementioned strong properties. In our simulations, on random instances of the size of the New York City school system, it terminates on average in less than three minutes (see Figure 1).

In this setting, schools are often perceived as commodities, and only students' welfare matters. Hence, enforcing stability implies a loss of efficiency. Abdulkadiroğlu et al. (2009) demonstrate the magnitude of

such efficiency loss with empirical data from the New York City school system, in which more than 4,000 eighth-graders in their sample could improve their assignments if stability constraints were relaxed. Striving to regain this loss in welfare for the students, many alternative concepts and mechanisms have, therefore, been introduced and extensively studied (see, e.g., Erdil and Ergin 2008, Kesten 2010, Abdulkadiroğlu et al. 2015, Kloosterman and Troyan 2016, Morrill 2016).

Those mechanisms lead to solutions outside the well-structured set of stable assignments. As a consequence, ad hoc structural studies and algorithms must be presented. Unfortunately, properties of the former and performance of the latter rarely match theory of and algorithms for stable assignments (Kesten 2010, Tang and Yu 2014, Kloosterman and Troyan 2016). For instance, Kesten's (2010) Efficiency Adjusted Deferred Acceptance Mechanism (EA-DAM) (one of the main focuses of the present paper), in our experiments, cannot terminate after 24 hours of computation, on average, on random instances of similar size as the New York City high school system. This algorithmic inefficiency harms the applicability of such mechanisms to real-world instances, especially if policy designers want to run them multiple times either as a subroutine in a more complex mechanism or to test the effects of different tie-breaking rules (Erdil and Ergin 2008, Ashlagi and Nikzad 2020).

The goal of this paper is to show how certain concepts, introduced in the literature to regain the loss of welfare caused by stability constraints, can be fully understood through the lens of classic theory of stable assignments. Moreover, we show that this better understanding leads to theoretically and practically faster algorithms as well as extensions and new connections within this classic theory. We believe that our results can stimulate further applications of those concepts as well as future theoretical research. The two topics that we study in depth are *legal assignments* (Morrill 2016) and *EADAM with consent* (Kesten 2010). Let us, therefore, introduce them next.

#### 1.1. Legal Assignments

Legality gives an alternative interpretation of fairness in an attempt to eliminate the tension between stability and efficiency. The stability condition prohibits, in the assignment chosen, the existence of a student–school pair that prefer each other to their assigned partners. Such pairs are called *blocking pairs*. Therefore, stability makes sure that no student is

harmed, and thus, no student has the justification to take legal action against the public school system. However, Morrill (2016) observes that legal standing, as interpreted by the U.S. Supreme Court, is not exactly the same as prohibiting blocking pairs. Specifically, in order for a student to have legal standing, not only must the student be harmed (i.e., forming a blocking pair with a school), this harm also must be *redressable*. That is, there must be an assignment that is accepted as feasible under which the student is assigned to the school.

With this interpretation, an institution is safe from legal actions if the set  $\mathcal{L}$  of assignments that are considered feasible has the property that if a student–school pair blocks an assignment from  $\mathcal{L}$ , then this pair is not matched in any assignment from  $\mathcal{L}$  (internal stability). On the other hand, in order to justify the exclusion of an assignment M from the set  $\mathcal{L}$ , there must be a pair that blocks M and is matched in some assignment from  $\mathcal{L}$  (external stability). Following Morrill (2016), we call a set  $\mathcal{L}$  with those properties legal. Note that every legal set contains the set of stable assignments. We illustrate this concept with an example.

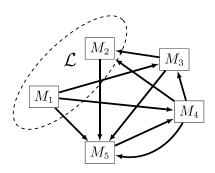
**Example 1.1.** Here and throughout the paper, one side of the bipartition is called *students* and the other is called *schools*. In this example, we also assume that each school can admit at most one student. Consider the instance with preference lists.

student 1:	A	В	C	school A:	2	3	1
student 2:	В	A		school B:	1	2	
student 3:	Α	C		school C:	3	1	

We list all five maximal matchings. Note that it is sufficient to consider only maximal matchings because if a matching is not maximal, it cannot be in a legal set. We also list the blocking pairs each matching admits. In this instance,  $M_1$  is the only stable matching.

#	maximal matching	blocking pairs
$\overline{M_1}$	1 <i>B</i> ,2 <i>A</i> ,3 <i>C</i>	Ø
$M_2$	1 <i>A</i> , 2 <i>B</i> , 3 <i>C</i>	3 <i>A</i>
$M_3$	1 <i>B</i> , 3 <i>A</i>	2 <i>A</i>
$M_4$	1 <i>C</i> , 2 <i>B</i> , 3 <i>A</i>	1 <i>B</i>
$M_5$	1 <i>C</i> ,2 <i>A</i>	1 <i>B</i> , 2 <i>B</i> , 3 <i>C</i>

We now construct a digraph with each maximal matching as a vertex. We add an arc (M, M') if and only if matching M blocks matching M', where we say a matching M blocks M' if M contains an edge that is a blocking pair for M'.



By the definition of legality, we claim that, in this instance, the set  $\mathcal{L} = \{M_1, M_2\}$  (circled) is a legal set because it satisfies both internal and external stability. This is because  $M_1$  and  $M_2$  do not block each other, and all other matchings are blocked by at least one of  $M_1$  and  $M_2$ . Although  $M_2$  is not stable, the harm resulting from blocking pair (3,A) is not redressable given that student 3 is not matched to school A in any matching from the set  $\mathcal{L}$ . Moreover, every student is weakly better off in  $M_2$  compared with in  $M_1$ .  $\diamondsuit$ 

From the example, one can see that relaxing stability to legality allows us to extend the set of feasible assignments while maintaining a certain level of fairness. As we show in Section 8, the increase in the size of the feasible set can be very significant. Morrill (2016) observes that, in the present setting, legality coincides with the concept of von Neumann–Morgenstern (vNM) stability (Von Neumann and Morgenstern 1953) in game theory under an appropriate definition of dominance. This is investigated in the one-to-one case by Wako (2010). (We defer a more detailed discussion of von Neumann-Morgenstern stability and of the work of Wako (2010) to Section 1.4). Morrill (2016) also shows that every one-to-many instance has a unique legal set  $\mathcal{L}$ . Moreover, assignments in  $\mathcal{L}$  form a lattice under the classic dominance relation. By standard arguments, this implies the existence of a student-optimal legal assignment, which Morrill (2016) shows is *Pareto efficient* for students and can be found using Kesten's (2010) EADAM.

### 1.2. EADAM with Consent

Recall that EADAM stands for efficiency adjusted deferred acceptance mechanism. As the name implies, it aims at regaining the efficiency lost because of stability constraints. EADAM again achieves efficiency improvement without creating legal concerns, and it does so by obtaining students' consent to allow for certain blocking pairs. More specifically, starting from the student-optimal stable assignment, EADAM iteratively asks for certain students' consent to allow the removal of certain schools from their preference lists and then reruns Gale–Shapley's algorithm. This

removes the possibility that certain student–school pairs act as blocking pairs. We defer a detailed description of the algorithm to Section 7 and illustrate here a number of properties (shown in Kesten 2010, Tang and Yu 2014) that make EADAM attractive for school choice.

If a student is asked to give consent, whether the student consents, the student's assignment does not change, and thus, no student has the incentive not to consent, and no student is harmed under EADAM. Moreover, EADAM outputs an assignment that is *constrained efficient*. That is, this assignment does not violate any nonconsenting students' priorities (i.e., no nonconsenting student is part of a blocking pair), but any other assignment that is weakly preferred by all students does. When all students consent, the output is therefore Pareto efficient.

Although EADAM is not strategy-proof (i.e., a student can misstate the student's preference list in order to be assigned to a better school), Kesten (2010) remarks that violation of strategy-proofness does not necessarily imply easy manipulability in practice (see, e.g., Roth and Peranson 1999) as agents usually do not have complete information about the preferences of other agents in the market and are, thus, unlikely to engage in potentially profitable strategic behaviors (Roth and Rothblum 1999). Kesten (2010) also proves that any mechanism that improves over the student-optimal stable mechanism either violates some non-consenting students' priority or is not strategy-proof.

Although both EADAM and legal assignments are further analyzed and extended by several authors (see, e.g., Kloosterman and Troyan 2016, Afacan et al. 2017, Dur et al. 2019, Ehlers and Morrill 2020, Tang and Zhang 2021), our knowledge of those two concepts is far from complete. In particular, the knowledge that legal assignments form a lattice gives little information on how to exploit it for algorithmic purposes, for example, how to find the legal assignment that maximizes some linear profit function. Moreover, little is known on how to exploit the structure of legal assignments to obtain the output of EADAM when not all students consent because the assignment output by the algorithm may not be legal.

#### 1.3. Our Contribution

Our first contribution addresses the structure of legal assignments. We prove in Section 3 that the set of legal assignments coincides with the set of stable assignments in a subinstance of the original one. That is, we can describe the set of legal assignments exactly as a set of stable assignments in a subinstance. The proof we present in Section 3 builds on results by Morrill (2016). However, a self-contained, not much harder proof that only relies on classic concepts from the theory of stable assignment is also possible, and it is

presented in E-companion EC.3. This latter proof studies the fixed points of a certain function defined in Morrill (2016), but our approach greatly simplifies the overall treatment. In particular, Morrill (2016) proves many properties of legal assignments from scratch and shows that the concept of legal assignments is well defined through its connection with EADAM. In our proof, we first show that legal assignments are stable in a subinstance. Afterward, all structural properties shown in Morrill (2016) follow immediately. We defer a more detailed comparison between those proofs to E-companion EC.3. We also show, by building on our approach and on results by Ehlers and Morrill (2020), that legal assignments coincide with the set of stable assignments in a subinstance for the more general case in which school preferences are represented by substitutable choice functions that satisfy the law of aggregate demand. We defer details to Section 3 and the Ecompanion.

As our second contribution, in Section 6, we show how to obtain the aforementioned subinstance in time linear in the number of edges of the input. Hence, in order to solve an optimization problem over the set of legal assignments (e.g., to find the already mentioned school optimal or other assignments of interest, such as the egalitarian, profit-optimal, minimum regret), one can resort to the broad literature on algorithms developed for the same problem on the set of stable assignments (see, e.g., Manlove 2013 for a collection of those results). Because the worst-case running time of those algorithms is at least linear in the number of edges, the complexity of the related problems over the set of legal assignments does not exceed their complexity over the set of stable assignments. To achieve this second contribution, we rely on the concept of metarotations (Bansal et al. 2007) and develop a symmetric pair of algorithms in Section 5, which we name student-rotate-remove and school-rotate-remove, that, respectively, find the school-optimal and studentoptimal legal assignments.

Our third contribution is a fast algorithm for EA-DAM with consent. Algorithmic results imply that, when all students consent, EADAM can be implemented as to run with the same time complexity as that of Gale–Shapley's, which is linear in the number of edges of the input. However, when only some students consent, the output of EADAM may no longer be legal. We show in Section 7 how to modify school-rotate-remove to produce the output of EADAM, again within the same time bound as Gale–Shapley. Hence, for two-sided matching markets, if one were to switch from the currently widely used deferred acceptance mechanism to EADAM, the computational time required to obtain a solution would not significantly increase. Computational tests on random instances

performed in Section 7.4 confirm that our algorithms run significantly faster in practice.

As our last contribution, we show that, when relaxing stability to legality, we can greatly increase the number of feasible matchings. We show one-to-one instances that have only one stable matching but exponentially many (in the number of agents) legal matchings. This is achieved by an exploration of the connection between *Latin marriages* introduced by Benjamin et al. (1995) and legal matchings. We defer the details to Section 8.

Our algorithm implementations for (1) finding student- and school-optimal legal assignments and obtaining the legal subinstance and (2) EADAM with consent can be found online.<sup>4</sup>

#### 1.4 Literature Review

There is a vast amount of literature on mechanism design for the school choice problem, balancing the focus among strategy-proofness, efficiency, and stability. From a theoretical perspective, Ergin (2002) shows that, under certain acyclicity conditions on the priority structure, the student-optimal stable assignment is also Pareto efficient for the students. Kesten (2010) interprets these cycles as sets of interrupting pairs (see Section 7.1 for a formal definition) and proposes EADAM, which improves efficiency by obtaining students' consent to waive their priorities.

Extending upon Kesten's (2010) framework, many researchers offer new perspectives. Tang and Yu (2014) propose a simplified version of EADAM, which repeatedly runs Gale–Shapley's algorithm after fixing the assignments of underdemanded schools. Bando (2014) shows an algorithm that iteratively runs Gale–Shapley's algorithm after fixing the assignments of the set of last proposers. Bando (2014) also shows that, when restricting to the one-to-one setting, his algorithm finds the student-optimal matching in the *vNM stable set*. The vNM stable set is a concept proposed by Von Neumann and Morgenstern (1953) for cooperative games. The definition of vNM stable set requires an irreflexive dominance relation among outcomes in the set.

For the stable assignment problem, the definition of legal assignments in Morrill (2016) corresponds to a vNM stable set under the dominance relation dom, where assignment  $M_1 \ dom \ M_2$  if  $M_1 \ blocks \ M_2$ . Under this dominance relation, results from Ehlers (2007) and Wako (2010) show existence and uniqueness of the vNM stable set in the one-to-one setting. Morrill (2016) further proves the existence and uniqueness results in the one-to-many setting as well as the fact that the vNM stable set has a lattice structure. Morrill (2016) is superseded by Ehlers and Morrill (2020), in which the concept of legality and the aforementioned results are generalized to the setting in which schools' preferences are

specified by substitutable choice functions that satisfy the law of aggregate demand. To the best of our understanding, results from Ehlers and Morrill (2020) do not have any implication in the stable assignment setting other than those that already follow from Morrill (2016) mentioned earlier. Interestingly, Ehlers and Morrill (2020) also investigate a different dominance relation dom' (which they call "vNM-blocks") and observe that dom and dom' lead to different vNM stable sets.

Wako (2010) presents an algorithm that finds the man- and woman-optimal matchings in the vNM stable set (under the dominance relation *dom* defined earlier) in the one-to-one case and shows that the vNM stable set coincides with the set of stable matchings in another instance. When restricted to the one-to-one case, our algorithms from Section 5 essentially project to that of Wako (2010) as Wako (2010) also obtains, for example, the woman-optimal legal matching by starting from the woman-optimal stable matching and iteratively finding rotations and eliminating edges. However, our approach is different because, unlike Wako (2010), we show that legal assignments are stable assignments in a subinstance before and independently of the algorithm for finding them. Even when restricted to the one-to-one case, this allows for a more direct derivation and, we believe, a more intuitive understanding of the algorithm and a simpler and shorter proof overall. Moreover, as Wako (2010) points out, his results have neither structural nor algorithmic implications for the vNM stable set in the one-to-many setting, and he actually poses as an open question to construct an algorithm to produce such assignments.

Our results answer this open question and allow us to also characterize legal assignments in the more general setting of Ehlers and Morrill (2020). We remark that, although there is a standard reduction from one-to-many instances to one-to-one instances (Gusfield and Irving 1989, Roth and Sotomayor 1992) such that the set of stable assignments of the former and the set of stable matchings of the latter correspond, this one-to-one mapping fails for the set of legal assignments (see Example EC.2.1). So we need to directly tackle the one-to-many setting.

#### 2. Basics

We introduce here basic notions and facts. We point readers to the book by Gusfield and Irving (1989) for a more comprehensive introduction on stable marriage and stable assignment problems.

For  $n \in \mathbb{N}$ , we denote by [n] the set  $\{1, ..., n\}$ . All (di)graphs in this paper are simple. All paths and cycles in the (di)graphs are, therefore, uniquely determined by the sequence of nodes they traverse and are denoted using this sequence, for example,

 $a_0,b_0,a_1,b_1,\cdots$ . The edge connecting two nodes a,b in an undirected graph is denoted by ab. For a graph G, we denote by V(G) and E(G) its sets of vertices and edges, respectively. For  $v\in V(G)$ , we let  $\deg_G(v)$  denote the degree of v (i.e., the number of adjacent vertices of v) in G. For a graph G(V,E) and  $F\subseteq E$ , we denote by G[F]:=G(V,F). A singleton of a graph is a node of degree zero. For sets  $S,S',S\triangle S'$  denotes their symmetric difference. A sink of a digraph is a node of outdegree zero.

An instance of the stable assignment problem is a triple  $(G, <, \mathbf{q})$  with  $G = (A \cup B, E)$ , where G is a bipartite graph with bipartition (A, B), < denotes the set  $\{<_v\}_{v\in A\cup B}$  with  $<_v$  being a strict ordering of the neighbors of v in G, and  $\mathbf{q} = \{q_b\}_{b \in B} \in \mathbb{N}^B$  denotes the maximum number of vertices in A that can be assigned to each  $b \in B$ . The number  $q_b$  is called the *quota* of b. Elements of A are referred to as students, and elements of *B* are referred to as schools. For  $x, y, y' \in A \cup B$  with  $xy, xy' \in E$ , we say x strictly prefers y to y' if y > xy', and we say that x (weakly) prefers y to y' and write  $y \ge_x y'$  if  $y >_x y'$  or y = y'. For all  $xy \in E$ , we assume  $y > x\emptyset$ . That is, the fact that ab is an edge in E means that *a* prefers to be assigned to *b* than to be unassigned and b prefers to accept a than to accept fewer than  $q_b$ students. When q is the vector all of ones, we speak of an instance of the stable marriage problem and denote it by (G, <). In this case, elements of A are referred to as men and elements of B are referred to as women.

An assignment M for an instance  $(G, <, \mathbf{q})$  is a collection of edges of *G* such that at most one edge of *M* is incident to a for each  $a \in A$ ; at most  $q_b$  edges of M are incident to *b* for each  $b \in B$ . For  $x \in A \cup B$ , we write  $M(x) = \{y : xy \in M\}$ . When  $M(x) = \{y\}$ , we often think of M(x) as an element instead of a set and write M(x)= y. For  $ab \in E$  and an assignment M, we call ab a blocking pair for *M* if student *a* prefers school *b* to the student's currently assigned school (i.e.,  $b >_a M(a)$ ) and school b either has empty seats (i.e.,  $|M(b)| < q_b$ ) or it prefers student a to someone who is currently occupying a seat at school b (i.e.,  $a>_b a'$  for some  $a' \in M(b)$ ). In this case, we say that ab blocks M, and similarly, we say that M' blocks M for every assignment M' containing edge ab. 5 An assignment is stable if it is not blocked by any edge of G.

We let  $\mathcal{M}(G, \mathbf{q})$  be the set of all assignments of  $(G, <, \mathbf{q})$  and let  $\mathcal{S}(G, <, \mathbf{q})$  be the set of all stable assignments of  $(G, <, \mathbf{q})$ . For a subgraph G' of G, we denote by  $(G', <, \mathbf{q})$  the stable assignment instance whose preference lists are those induced by < on G', and quotas are those obtained by restricting  $\mathbf{q}$  to nodes in G'. We say that a student–school pair (a, b) is a *stable pair* if there exists a stable matching  $M \in \mathcal{S}(G, <, \mathbf{q})$  in which student a is assigned to school

*b*. In such cases, we also say that *a* is a *stable partner* of *b*.

Every instance has at least one stable assignment. Algorithms proposed by Gale and Shapley (1962) output special stable assignments. The following theorem collects results from Gale and Shapley (1962) and Gusfield and Irving (1989).

**Theorem 2.1.** The student-proposing Gale–Shapley's algorithm outputs a stable assignment  $M_0$  that is optimal for the students: for any stable assignment  $M \in \mathcal{S}(G, <, \mathbf{q})$ , every student a prefers  $M_0$  to M (i.e.,  $M_0(a) \geq_a M(a)$ ). Similarly, the school-proposing Gale–Shapley's algorithm outputs a stable assignment  $M_z$  that is optimal for the schools. Moreover,  $M_z$  is student-pessimal: for any stable assignment  $M \in \mathcal{S}(G, <, \mathbf{q})$ , every student a prefers M to  $M_z$  (i.e.,  $M(a) \geq_a M_z(a)$ ).

# 3. Legal Assignments Are Stable Assignments in Disguise

For an instance  $(G, <, \mathbf{q})$  of the stable assignment problem and a set of assignments  $\mathcal{M}' \subseteq \mathcal{M}(G, \mathbf{q})$ , define  $\mathcal{I}(\mathcal{M}')$  as the set of assignments that are blocked by some assignment from  $\mathcal{M}'$ . We say a set of assignments  $\mathcal{M}'$  has the *legal property* if no assignment from  $\mathcal{M}'$  is blocked by any assignment from  $\mathcal{M}'$  (internal stability), and every assignment not in  $\mathcal{M}'$  is blocked by some assignment from  $\mathcal{M}'$  (external stability). These two requirements can be summarized as a fixed-point condition:  $\mathcal{I}(\mathcal{M}') = \mathcal{M}(G, \mathbf{q}) \setminus \mathcal{M}'$ . In this case, we say that  $(\mathcal{M}', \mathcal{I}(\mathcal{M}'))$  is a *legal partition* of  $\mathcal{M}(G, \mathbf{q})$ .

We devote this section to the proof of the following theorem.

**Theorem 3.1.** Let  $(G, <, \mathbf{q})$  be an instance of the stable assignment problem. There exists a unique set of assignments  $\mathcal{L} \subseteq \mathcal{M}(G, \mathbf{q})$  that has the legal property. This set coincides with the set of stable assignments in  $(G_L, <, \mathbf{q})$ , where  $G_L$  is a subgraph of G induced by all and only edges that are in some assignment from  $\mathcal{L}$ . That is,

$$E(G_L) = \bigcup \{M : M \in \mathcal{S}(G_L, <, \mathbf{q})\} = \bigcup \{M : M \in \mathcal{L}\}.$$

As observed in Gusfield and Irving (1989) and Roth and Sotomayor (1992), there is a one-to-one correspondence between stable assignments in  $(G, <, \mathbf{q})$  and stable matchings in a reduced stable marriage instance  $(H_G, <_G)$  (for a detailed description of the reduction, see E-companion EC.1). One could think of proving Theorem 3.1 by showing the (simpler) results for the instance  $(H_G, <_G)$  and then deducing the set of legal matchings of  $(H_G, <_G)$ . Unfortunately, the bijection between stable assignments and stable matchings does

not extend to the legal setting; see E-companion EC.2 for an example.

The proof of Theorem 3.1 presented in this section relies on the following result by Morrill (2016).

**Theorem 3.2.** Let  $(G, <, \mathbf{q})$  be an instance of the stable assignment problem. There exists a unique set of assignments  $\mathcal{L} \subseteq \mathcal{M}(G, \mathbf{q})$  that satisfies the legal property.

As the proof of Theorem 3.2 from Morrill (2016) is somehow involved, in E-companion EC.3, we present an alternative proof of Theorem 3.1 not relying on Theorem 3.2.

For a stable assignment instance  $(G, <, \mathbf{q})$ , we denote by  $\mathcal{L}(G, <, \mathbf{q}) \subseteq \mathcal{M}(G, <, \mathbf{q})$  the unique set of assignments that satisfies the legal property. Moreover, we call  $\mathcal{L}(G, <, \mathbf{q})$  the set of legal assignments of instance  $(G, <, \mathbf{q})$ . We say that an edge  $e \in E(G)$  is legal if it is contained in some assignment from  $\mathcal{L}(G, <, \mathbf{q})$  and is illegal otherwise.

For the rest of the section, we fix a stable assignment instance  $(G, <, \mathbf{q})$  with  $G = (A \cup B, E)$  and let  $\mathcal{M} := \mathcal{M}(G, \mathbf{q})$  be the set of assignments,  $\mathcal{L} := \mathcal{L}(G, <, \mathbf{q})$  be the set of legal assignments,  $\overline{\mathcal{L}} := \mathcal{M} \setminus \mathcal{L}$  be the set of illegal assignments,  $\overline{\mathcal{E}} := \bigcup \{M : M \in \mathcal{L}\}$  be the set of legal edges, and  $G_L := G[\overline{E}]$  be the subgraph with the illegal edges removed. The next lemmas show that illegal edges can be removed from G without modifying the set of legal assignments.

**Lemma 3.1.** Let e be an illegal edge. Removing the edge e from the instance does not change the set of legal assignments. That is,  $\mathcal{L} = \mathcal{L}(\widetilde{G}, <, \mathbf{q})$ , where  $\widetilde{G} := G[E \setminus \{e\}]$ .

**Proof.** Let  $\mathcal{M}^e := \{M \in \mathcal{M} : e \in M\}$  be the set of assignments that contain the illegal edge e and let  $\mathcal{M}$  :=  $\mathcal{M}(G, \mathbf{q})$  be the set of all assignments of the instance  $(G, <, \mathbf{q})$ . Note that  $\mathcal{M} = \mathcal{M} \setminus \mathcal{M}^e$ . Moreover, all legal assignments are assignments of the instance  $(G, <, \mathbf{q})$ because *e* is an illegal edge:  $\mathcal{L} \subseteq \mathcal{M}$ . Hence,  $(\mathcal{L}, \mathcal{M} \setminus \mathcal{L})$ is a partition of  $\mathcal{M}$ . We show next that it is also a legal partition. To see this, first note that any two assignments  $M_1, M_2 \in \mathcal{L}$  do not block each other because  $\mathcal{L}$  is the set of legal assignments of the original instance. Next, consider any assignment  $M' \in \mathcal{M} \setminus \mathcal{L}$ . Then, M' is an illegal assignment in the original instance and must be blocked by some assignment in  $\mathcal{L}$ . Thus, together with the uniqueness of the legal partition given by Theorem 3.2, we conclude that  $\mathcal{L}(G, <, \mathbf{q}) = \mathcal{L}$ .  $\square$ 

**Lemma 3.2.** The set of legal assignments does not change after removing all illegal edges. That is,  $\mathcal{L} = \mathcal{L}(G_L, <, \mathbf{q})$ .

**Proof.** Let  $e_1, e_2, \dots, e_k$  be an ordering of the illegal edges, and for  $i = 1, 2, \dots, k$ , let  $G^i := G[E \setminus \{e_1, e_2, \dots, e_i\}]$  be a sequence of subgraphs obtained after

successively removing illegal edges. Observe that  $G^k = G_L$ . By Lemma 3.1, we have  $\mathcal{L}(G^1, <, \mathbf{q}) = \mathcal{L}$ , and thus, by definition of illegal edges, edges  $e_2, \cdots, e_k$  remain illegal in instance  $(G^1, <, \mathbf{q})$ . Therefore, applying Lemma 3.1 again to  $(G^1, <, \mathbf{q})$ , we have  $\mathcal{L}(G^2, <, \mathbf{q}) = \mathcal{L}(G^1, <, \mathbf{q}) = \mathcal{L}$ . Iterating the process, we can conclude that  $\mathcal{L}(G_L, <, \mathbf{q}) = \mathcal{L}(G^k, <, \mathbf{q}) = \mathcal{L}(G^{k-1}, <, \mathbf{q}) = \dots = \mathcal{L}$ .  $\square$ 

**Lemma 3.3.** Once all illegal edges have been removed, the set of stable assignments coincides with the set of legal assignments:  $S(G_L, <, \mathbf{q}) = \mathcal{L}(G_L, <, \mathbf{q})$ .

**Proof.** The direction  $S(G_L, <, \mathbf{q}) \subseteq \mathcal{L}(G_L, <, \mathbf{q})$  is clear because a stable assignment is not blocked by any other assignment. For the other direction, let  $M \in \mathcal{L}(G_L, <, \mathbf{q})$  be a legal assignment of the instance  $(G_L, <, \mathbf{q})$ . Then, M is not blocked by any assignment in  $\mathcal{L}(G_L, <, \mathbf{q})$  because of internal stability. Because every edge in  $E(G_L)$  appears in at least one assignment in  $\mathcal{L}(G_L, <, \mathbf{q})$ , M admits no blocking pair in  $G_L$  and, thus, is stable in the instance  $(G_L, <, \mathbf{q})$ . This concludes the proof.  $\square$ 

**Proof of Theorem 3.1.** This follows immediately from Theorem 3.2 and Lemmas 3.2 and 3.3.  $\Box$ 

The approach developed in this section can be extended to the more general setting studied in Ehlers and Morrill (2020), in which schools' preferences are represented by certain choice functions. In particular, Theorem 3.1 also holds in this setting. We defer the details to E-companion EC.4.

We show that legal assignments are stable assignments in  $G_L$ . Because there might be an exponential number of legal assignments, one cannot expect to construct  $G_L$  efficiently by explicitly listing all the legal assignments. Instead, the main tool we use is an efficient mechanism in identifying legal and illegal edges, which is developed in Section 5.

Before we introduce this algorithm, we need some more properties of stable assignments.

## 4. The Structure of Stable Assignments

In this section, we recall known results on structural properties of stable assignments and their algorithmic consequences. Throughout the section, we fix a stable assignment instance  $(G, <, \mathbf{q})$  with  $G = (A \cup B, E)$ . Given two assignments  $M, M' \in \mathcal{M}(G, \mathbf{q})$ , we say M (weakly) dominates M' and write  $M \geq M'$  if every student a prefers M to M':  $M(a) \geq_a M'(a)$ ,  $\forall a \in A$ . If, moreover,  $M \neq M'$ , we say that M strictly dominates M' and write M > M'. An assignment  $M \in \mathcal{M}(G, \mathbf{q})$  is said to be Pareto efficient (for students) if there is no other assignment  $M' \in \mathcal{M}(G, \mathbf{q})$  such

that M' dominates M. The following fact is well known (see, e.g., Gusfield and Irving 1989).

**Theorem 4.1.** The set of stable assignments  $S(G, <, \mathbf{q})$  endowed with the dominance relation  $\succeq$  forms a distributive lattice. In particular, there exist stable assignments  $M_0$  and  $M_z$  such that  $M_0 \succeq M \succeq M_z$  for all stable assignment  $M \in S(G, <, \mathbf{q})$  (it is possible that  $M_0 = M_z$ ).  $M_0$  and  $M_z$  are called the student- and school-optimal stable assignments, respectively.

Note that the student-optimal (respectively, school-optimal) stable assignment coincides with the one output by Gale–Shapley's algorithm with students (schools) proposing as described in Theorem 2.1. Hence, the notation describing those assignments coincide.

Next, we introduce the concept of rotations in the one-to-many setting. Informally speaking, a rotation exposed in a stable assignment M is a certain M-alternating cycle C such that  $M \triangle C$  is again a stable assignment. C has the property that every agent from one side of the bipartition prefers M to  $M \triangle C$ , and every agent from the other side prefers  $M \triangle C$  to M. We can interpret a rotation as a cycle of unmatches and rematches with one side getting better and the other side getting worse. Hence, rotations provide a mechanism to generate one stable assignment from another, moving along the distributive lattice formed by the set of stable assignments.

Because of the different role played by the two sides of the bipartition, we distinguish between *school*- and *student-rotations*. In the following, we present them jointly by choosing X to be one side of the bipartition and Y the other. We extend the quota vector  $\mathbf{q}$  to students by letting  $q_a = 1$  for each student  $a \in A$ .

For a stable assignment  $M \in \mathcal{S}(G, <, \mathbf{q})$  and agent  $x \in X$ , let  $s_M(x)$  be the first agent  $y \notin M(x)$  on x's preference list such that y prefers x to one of y's partners (i.e.,  $x >_y x'$  for some  $x' \in M(y)$ ). If  $y := s_M(x)$  exists, we must have that x prefers all of x's partners over agent y (i.e.,  $y' >_x y$  for all  $y' \in M(x)$ ) because M is stable. If, moreover,  $|M(y)| = q_y$ , define  $next_M(x)$  to be the least preferred partner of agent y among all current partners of y. That is,  $next_M(x) \in M(y)$ , and for all  $x' \in M(y)$ ,  $x' \geq_y next_M(x)$ . If, otherwise,  $|M(y)| < q_y$ , then define  $next_M(x) = \emptyset$ .

Given distinct  $x_0, \ldots, x_{r-1} \in X$  and  $y_0, \ldots, y_{r-1} \in Y$ , a cycle  $y_0, x_0, y_1, x_1, \ldots, y_{r-1}, x_{r-1}, y_0$  of G is an X-rotation exposed in M if  $s_M(x_i) = y_{i+1}$  and  $next_M(x_i) = x_{i+1}$  for all  $i = 0, \ldots, r-1$  (here and later, indices are taken modulo r). Note that  $x_i y_i \in M$  for all  $i = 0, \ldots, r-1$ . Let  $D_X$  be the digraph with vertices  $X \cup Y \cup \{\emptyset\}$ , and arcs (x, y) and (y, x') if and only if  $s_M(x) = y$  and  $next_M(x) = x'$ .

We call  $D_X$  the *X-rotation digraph* (of M) and denote by  $A(D_X)$  the set of arcs of  $D_X$ . If  $s_M(x)$  does not exist for some agent  $x \in X$ , then x is a sink in  $D_X$ . Thus, note that sinks in  $D_X$  are either agents in X or  $\emptyset$ . One easily observes that X-rotations exposed in M are in one-to-one correspondence with directed cycles in  $D_X$ .

Let  $\rho := y_0, x_0, \cdots, y_{r-1}, x_{r-1}$  be an X-rotation exposed in M. The *elimination* of  $\rho$  maps stable assignment M to the assignment  $M' := M/\rho$ , where M'(x) = M(x) for every agent x who is not in the rotation (i.e.,  $x \in X \setminus \rho$ ) and  $M'(x_i) = (M(x_i) \setminus \{y_i\}) \cup \{y_{i+1}\}$  for  $i = 0, 1, \cdots, r-1$ . Note that the mapping is well defined because it is easy to check that M' is an assignment.

An *X*-rotation (digraph) is called a student or school rotation (digraph), respectively, when *X* is the set of students or schools. When it is clear whether we are referring to students or schools, we drop the prefix. See Example 5.1 for an illustration of rotations and rotation digraphs in the context of our algorithm.

The following lemmas (Bansal et al. 2007) extend classic results on rotations in the one-to-one setting to our one-to-many setting. They show that the set of stable assignments is complete and closed under the elimination of exposed rotations.

**Lemma 4.1.** Let  $M \in \mathcal{S}(G, <, \mathbf{q})$  be a stable assignment,  $\rho$  be an X-rotation exposed in M, and  $M' = M/\rho$  be the assignment obtained after eliminating  $\rho$  from M. Then, M' is stable in  $(G, <, \mathbf{q})$  (i.e.,  $M' \in \mathcal{S}(G, <, \mathbf{q})$ ). Moreover, M strictly dominates M' (i.e., M > M') if X is the set of students and M is strictly dominated by M' (i.e., M' > M) if X is the set of schools. If there is no X-rotation exposed in M, M is the Y-optimal stable assignment. In addition, every stable assignment can be generated by a sequence of X-rotation eliminations, starting from the X-optimal stable assignment, and every such sequence contains the same set of X-rotations.

**Lemma 4.2.**  $xy \in E$  is a stable pair if and only if (i) either x is assigned to y in the Y-optimal stable assignment or (ii) for some X-rotation  $y_0, x_0, y_1, x_1, \dots, y_{r-1}, x_{r-1}$  exposed in some stable assignment, we have  $x = x_i$  and  $y = y_i$  for some  $i \in \{0, \dots, r-1\}$ .

We refer to Baïou and Balinski (2004) for further results on the stable assignment model.

For an instance  $(G, <, \mathbf{q})$ , we denote by  $\mathcal{R}(G, <, \mathbf{q})$  the set of student rotations exposed in some of its stable assignments and by  $\mathcal{SR}(G, <, \mathbf{q})$  the set of school rotations exposed in some of its stable assignments.

**Lemma 4.3.** We have:  $|\mathcal{R}(G, <, \mathbf{q})| = |\mathcal{SR}(G, <, \mathbf{q})|$ . There is a bijection  $\sigma : \mathcal{R}(G, <, \mathbf{q}) \to \mathcal{SR}(G, <, \mathbf{q})$  between the set of student rotations and the set of school rotations such that, for each stable assignment  $M \in \mathcal{S}(G, <, \mathbf{q})$  and student rotation  $\rho \in \mathcal{R}(G, <, \mathbf{q})$  exposed in M, we have  $M = (M/\rho)/\sigma(\rho)$ .

# 5. Algorithms for Student- and School-Optimal Legal Assignments

**Algorithm 5.1** (*X*-Rotate-Remove to Find the *Y*-Optimal Legal Assignment)

**Require:**  $(G(A \cup B, E), <, \mathbf{q})$ 

- 1. Find the Y-optimal stable assignment  $M_Y$  of  $(G, <, \mathbf{q})$  via Gale–Shapley's algorithm.
- 2. Let  $G^0 := G$  and  $M^0 := M_Y$ .
- 3. Set i = 0 and let  $D^0$  to be the X-rotation digraph of  $M^0$  in  $(G^0, <, \mathbf{q})$ .
- 4. **while**  $D^i$  still has an arc, **do**
- 5. Find (i) arcs  $(x', y), (y, x) \in A(D^i)$ , where x is a sink in  $D^i$  or (ii) a cycle  $C^i$  of  $D^i$ .
- 6. **if** (i) is found, **then**
- 7. Define  $G^{i+1}$  from  $G^i$  by removing x'y, and set  $M^{i+1} = M^i$ .
- 8. **else if** (ii) is found, **then**
- 9. Let  $\rho^i$  be the corresponding *X*-rotation. Set  $M^{i+1} = M^i/\rho^i$ , and  $G^{i+1} = G^i$ .
- 10. **end if**
- 11. Set i = i + 1 and let  $D^i$  be the X-rotation digraph of  $M^i$  in  $(G^i, <, \mathbf{q})$ .
- 12. end while
- 13. Output  $M^i$ .

Because of Theorems 3.1 and 4.1, the concepts of student- and school-optimal legal assignments are well defined. In this section, we show efficient routines for finding them. Throughout the section, we again fix a stable assignment instance  $(G, <, \mathbf{q})$  with  $G = (A \cup B, E)$ . We denote by  $M_0^{\mathcal{L}}$  and  $M_z^{\mathcal{L}}$  the student-and school-optimal legal assignments, respectively.

Suppose first that we want to find the student-optimal legal assignment. The basic idea of the algorithm is the following: at each iteration, a legal assignment M and a set of edges identified as illegal are taken as input, and one of the following three cases happens: either (i) the set of edges identified as illegal is expanded, (ii) a legal assignment M' that strictly dominates M (i.e., M' > M) is produced, or (iii) M is certified as the student-optimal legal assignment. If we are in case (i), then we can safely remove the newly found illegal edge (because of Lemma 3.1) and proceed to the next iteration. If we are in case (ii), we replace M with M', and proceed to the next iteration. If we are in case (iii), we halt the algorithm and output the current assignment.

In order to distinguish between cases (i), (ii), and (iii), we rely on properties of the rotation digraph. In the following, *X* can again be either the set of students or the set of schools. The proof of the following lemma uses some of the tools developed in the proof of Theorem 3.1 given in the E-companion and, thus, is deferred to E-companion EC.5.

**Lemma 5.1.** Let  $M \in \mathcal{S}(G, <, \mathbf{q})$  be a stable assignment. If  $x \in X \cup \{\emptyset\}$  is a sink in the X-rotation digraph  $D_X$  of M

and  $(x',y),(y,x) \in A(D_X)$  for some  $x' \in X$  and  $y \in Y$ , then x'y is an illegal edge.

Hence, if the algorithm finds a sink fulfilling the properties of Lemma 5.1 in the school-rotation digraph, we are in case (i). If the school-rotation digraph has a directed cycle, eliminating the corresponding school rotation from M brings us to case (ii). Finally, if  $D_B$  has no arc, we conclude that we are in case (iii). The initial iteration starts with the set of identified illegal edges being empty and M being the student-optimal stable assignment. The algorithm that finds the school-optimal legal assignment proceeds similarly with a legal assignment M' that is dominated by M (i.e., M > M') generated in case (ii).

A formal description of our algorithm is given in Algorithm 5.1. Its correctness is shown in the proof of Theorem 5.1. We illustrate the algorithm in Example 5.1.

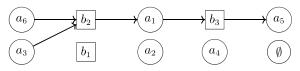
**Example 5.1.** We apply student- and school-rotate-remove to the following instance with six students and three schools, in which each school has a quota of two. In this and all following examples, when it is clear to whose preference list we are referring, the subscript in > is dropped.

$$\begin{array}{lll} a_1: \boxed{b_2} > b_3 > b_1 & b_1: a_1 > \boxed{a_4} > \boxed{a_3} > a_5 > a_2 > a_6 \\ \\ a_2: b_1 > \boxed{b_2} > b_3 & b_2: a_3 > \boxed{a_2} > a_6 > \boxed{a_1} > a_5 > a_4 \\ \\ a_3: b_3 > \boxed{b_1} > b_2 & b_3: \boxed{a_6} > a_1 > \boxed{a_5} > a_2 > a_4 > a_3 \\ \\ a_4: \boxed{b_1} > b_2 > b_3 & \\ \\ a_5: \boxed{b_3} > b_2 > b_1 & \\ \\ a_6: b_1 > \boxed{b_3} > b_2 & \\ \end{array}$$

The student- and school-optimal stable assignments coincide and are given by  $\{a_1b_2, a_2b_2, a_3b_1, a_4b_1, a_5b_3, a_6b_3\}$  (squared entries). This is the  $M^0$  for both algorithms.

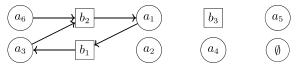
#### Student-Rotate-Remove

On  $a_1$ 's preference list,  $b_3$  is the first school after  $M^0(a_1)$ . In addition,  $b_3$  prefers  $a_1$  to  $a_5$ , who is  $b_3$ 's least preferred student among  $M^0(b_3)$ . Thus,  $s_{M^0}(a_1) = b_3$  and  $next_{M^0}(a_1) = a_5$ . After working out  $s_{M^0}(\cdot)$  and  $next_{M^0}(\cdot)$  of all the students, we have the rotation digraph  $D^0$  for the first iteration of student-rotate-remove.



Here, we find a case (i) with  $x' = a_1$ ,  $y = b_3$ , and  $x = a_5$ . So we set  $M^1 = M^0$ , remove  $x'y = a_1b_3$  from the

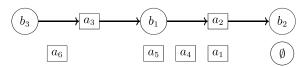
instance, and update the rotation digraph  $D^1$  for the next iteration.



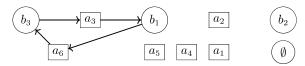
Now, we have a case (ii) with the corresponding student rotation  $\rho^1=b_2,a_1,b_1,a_3$ . Eliminating  $\rho^1$  from  $M^1$ , we have  $M^2=M^1/\rho^1=\{a_1b_1,a_2b_2,a_3b_2,\ a_4b_1,a_5b_3,\ a_6b_3\}$ . In the next iteration, the rotation digraph  $D^2$  only contains sinks. Thus, the algorithm terminates and outputs  $M^2$  as the school-optimal legal assignment.

#### School-Rotate-Remove

The first student on  $b_1$ 's preference list that prefers  $b_1$  to the student's assigned school under  $M^0$  is  $a_2$ . Thus,  $s_{M^0}(b_1) = a_2$  and  $next_{M^0}(b_1) = b_2$ . After working out  $s_{M^0}(\cdot)$  and  $next_{M^0}(\cdot)$  of all the schools, we have the rotation digraph  $D^0$  for the first iteration.



Here, we find a case (i) with  $x' = b_1$ ,  $y = a_2$ , and  $x = b_2$ . So we set  $M^1 = M^0$ , remove  $x'y = a_2b_1$  from the instance, and update the rotation digraph  $D^1$  for the next iteration.



Now, we have a case (ii) with the corresponding school rotation  $\rho^1 = a_6, b_3, a_3, b_1$ . Eliminating  $\rho^1$  from  $M^1$ , we have  $M^2 = M^1/\rho^1 = \{a_1b_2, a_2b_2, a_3b_3, a_4b_1, a_5b_3, a_6b_1\}$ . In the next iteration, the rotation digraph  $D^2$  only contains sinks. Thus, the algorithm terminates and outputs  $M^2$  as the student-optimal legal assignment.  $\square$ 

#### 5.1. Correctness of Algorithm 5.1

Using the lattice structure of the legal assignments, we can deduce the correctness of Algorithm 5.1.

**Theorem 5.1.** Algorithm 5.1 finds the Y-optimal legal assignment.

**Proof.** We focus on the statement with Y being the set of students; the other follows analogously. We first show, by induction on the iteration i of the algorithm, that  $M^i \in \mathcal{S}(G^i, <, \mathbf{q})$  and  $\mathcal{L}(G^i, <, \mathbf{q}) = \mathcal{L}$ . This is obvious for i = 0. Assume the claim is true for  $i - 1 \ge 0$  and consider iteration i. If the condition at step 6 is satisfied,  $M^i = M^{i-1}$  is unchanged, and the edge removed from  $G^{i-1}$  is illegal by Lemma 5.1. Hence,  $M^i = M^{i-1}$ 

 $M^{i-1} \in \mathcal{S}(G^{i-1}, <, \mathbf{q}) \subseteq \mathcal{S}(G^i, <, \mathbf{q})$  and  $\mathcal{L}(G^i, <, \mathbf{q}) = \mathcal{L}(G^{i-1}, <, \mathbf{q}) = \mathcal{L}$  by induction and Lemma 3.1. If, conversely, the condition at step 8 is satisfied, then  $\rho^{i-1}$  is a school rotation exposed in  $M^{i-1}$ , and  $M^i = M^{i-1}/\rho^{i-1} \in \mathcal{S}(G^{i-1}, <, \mathbf{q})$  by induction and Lemma 4.1. Moreover, because  $G^i = G^{i-1}$ , we have  $\mathcal{S}(G^i, <, \mathbf{q}) = \mathcal{S}(G^{i-1}, <, \mathbf{q})$  and  $\mathcal{L}(G^i, <, \mathbf{q}) = \mathcal{L}(G^{i-1}, <, \mathbf{q}) = \mathcal{L}$ .

In order to conclude the proof, observe that, at the end of the algorithm, the school-rotation digraph—call it  $D^*$ —only has sinks. We first claim that the assignment output—call it  $M^*$ —strictly dominates every assignment in  $\mathcal{M}(G^*,\mathbf{q})$ , where  $G^*$  is the graph at the end of the algorithm. Assume by contradiction that there is  $M\in \mathcal{M}(G^*,\mathbf{q})$  and a student a such that  $b:=M(a)>_a M^*(a)$ . Then,  $s_{M^*}(b)$  exists by definition, contradicting the fact that b is a sink in  $D^*$  (it is possible that  $s_{M^*}(b) \neq a$  as there may be other nodes that precede a in b's list and have the required property, but it is a contradiction regardless). By what we proved earlier, we know that  $\mathcal{L} = \mathcal{L}(G^*, <,\mathbf{q}) \subseteq \mathcal{M}(G^*,\mathbf{q})$ . By Theorems 3.1 and 4.1, legal assignments form a lattice with respect to the partial order  $\succeq$ . Hence,  $M^*$  is the student-optimal legal assignment.  $\square$ 

Note that the previous theorem in particular implies that the output of Algorithm 5.1 is unique regardless of how we choose between steps 6 and 8 at each iteration when multiple possibilities are present.

#### 5.2. Time Complexity

A straightforward implementation of Algorithm 5.1 requires the construction of a rotation digraph at each iteration. However, this is computationally expensive. Instead of obtaining the complete rotation digraph at each iteration, we only *locally* build and update a directed path of the rotation digraph until a cycle or a sink is found. Together with suitable data structures, we can achieve the time complexity of O(|E|).

**Theorem 5.2.** Algorithm 5.1 can be implemented so as to run in time O(|E|).

The full details of our implementation and the proof of Theorem 5.2 are included in E-companion EC.6.1 and EC.6.2.

# 6. An O(|E|) Algorithm for Computing $G_L$

Throughout this section, we fix an instance  $(G, <, \mathbf{q})$  with  $G = (A \cup B, E)$  and abbreviate the set of stable assignments as  $\mathcal{S} := \mathcal{S}(G, <, \mathbf{q})$ . We start with a preliminary fact. Recall that we denote by  $\mathcal{R}(G, <, \mathbf{q})$  and  $\mathcal{S}\mathcal{R}(G, <, \mathbf{q})$  the set of student and school rotations exposed in some stable assignment of  $(G, <, \mathbf{q})$ , respectively. Let  $G_L$  be the subgraph of G that includes all and only edges in some legal assignments in  $\mathcal{L}(G, <, \mathbf{q})$  as defined in Theorem 3.1.

**Lemma 6.1.** Let e be an illegal edge of  $(G, <, \mathbf{q})$  and let  $\widetilde{G} = G[E \setminus \{e\}]$ . Deleting edge e does not remove any

element from either the set of student rotations or the set of school rotations:  $\mathcal{R}(G, <, \mathbf{q}) \subseteq \mathcal{R}(\widetilde{G}, <, \mathbf{q})$  and  $\mathcal{S}\mathcal{R}(G, <, \mathbf{q}) \subseteq \mathcal{S}\mathcal{R}(\widetilde{G}, <, \mathbf{q})$ .

**Proof.** Fix a stable assignment  $M \in \mathcal{S}$ . Because  $\mathcal{S} \subseteq \mathcal{S}(\widetilde{G}, <, \mathbf{q})$ , M is also a stable assignment of  $(\widetilde{G}, <, \mathbf{q})$ . First, consider any student rotation  $\rho \in \mathcal{R}(G, <, \mathbf{q})$  exposed in M. We want to show that  $\rho$  is also exposed in M in  $(\widetilde{G}, <, \mathbf{q})$ . Assume  $\rho = b_0, a_0, b_1, a_1, \cdots, b_{r-1}, a_{r-1}$ . By Lemma 4.2, edges  $a_ib_{i+1}$  and  $a_{i+1}b_{i+1}$  for all  $i=0,1,\cdots,r-1$  are stable and, therefore, legal. Hence, all such edges are in  $E(\widetilde{G})$ , implying that  $b_{i+1} = s_M(a_i)$  and  $next_M(a_i) = a_{i+1}$  hold in  $(G, <, \mathbf{q})$  as well. Thus,  $\rho$  is exposed in M in  $(\widetilde{G}, <, \mathbf{q})$  and as desired. Therefore,  $\rho \in \mathcal{R}(\widetilde{G}, <, \mathbf{q})$  and  $\mathcal{R}(G, <, \mathbf{q}) \subseteq \mathcal{R}(\widetilde{G}, <, \mathbf{q})$ . A similar argument shows  $\mathcal{SR}(G, <, \mathbf{q}) \subseteq \mathcal{SR}(\widetilde{G}, <, \mathbf{q})$ .

**Theorem 6.1.** The subgraph  $G_L$  can be found in time O(|E|).

**Proof.** By Theorem 3.1 and Lemma 4.2,  $E(G_L)$  is given by all and only edges in the student-optimal legal assignment  $M_0^{\mathcal{L}}$ , plus all pairs  $a_i b_{i+1}$  for some student rotation  $\rho = b_0, a_0, \dots, a_k \in \mathcal{R}(G_L, <, \mathbf{q})$ . By Lemma 4.1, there exists exactly one set  $\mathcal{R}_1$  of student rotations whose elimination leads from  $M_0^{\mathcal{L}}$  to the studentoptimal stable assignment  $M_0$ ; one set  $\mathcal{R}_2$  leading from  $M_0$  to the school-optimal stable assignment  $M_z$  and one set  $\mathcal{R}_3$  leading from  $M_z$  to the school-optimal legal assignment  $M_7^{\mathcal{L}}$ . By Lemma 4.1,  $\mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3$  is the set of all rotations  $\mathcal{R}(G_L, <, \mathbf{q})$  of  $(G_L, <, \mathbf{q})$ . We argue that  $\mathcal{R}_3$  is computed during the execution of studentrotate-remove. Indeed, throughout the algorithm, a sequence of rotations is found and eliminated, leading from  $M_z$  to  $M_z^{\mathcal{L}}$ . Each of these is exposed in some stable assignment in an instance that contains all legal edges. Hence, by repeated application of Lemma 6.1, those rotations form set  $\mathcal{R}_3$ . They can be computed in time O(|E|) by Theorem 5.2. By Lemma 4.1 and repeated applications of Lemma 6.1,  $\mathcal{R}_2$  coincides with the set  $\mathcal{R}(G, <, \mathbf{q})$ , which can be computed in time O(|E|) by classical algorithms; see, for example, Gusfield and Irving (1989). School-rotateremove computes in time O(|E|), again by Theorem 5.2, the set of school rotations  $SR_1$  whose sequential elimination starting from  $M_0$  leads to  $M_0^{\mathcal{L}}$ . By Lemma 4.3, the set  $\mathcal{R}_1$  can be obtained from  $\mathcal{SR}_1$ via the bijection  $\sigma$ . Consider a student rotation  $\rho = b_0, a_0, b_1, a_1, \cdots, a_{r-1} \in \mathcal{R}_1$ . Because its corresponding school rotation  $\sigma(\rho) \in \mathcal{SR}_1$  can be obtained simply as  $a_0, b_1, a_1, \dots, a_{r-1}, b_0$ , computing  $\mathcal{R}_1$  from  $\mathcal{SR}_1$ takes time O(|E|). This concludes the proof.  $\Box$ 

# 7. An O(|E|) Algorithm for EADAM with Consent

In this section, we first formally introduce EADAM with consent (Kesten 2010). Then, in Section 7.2, we

show that a fast implementation of EADAM can be achieved by a suitable modification of our school-rotate-remove algorithm. The proof relies on a simplified and outcome-equivalent version of EADAM introduced by Tang and Yu (2014). Thus, we defer the proof as well as a formal introduction of simplified EADAM to E-companion EC.7. Together with Theorem 5.2, this implies the following.

**Theorem 7.1.** *EADAM with consent on a stable assignment instance*  $(G(A \cup B, E), <, \mathbf{q})$  *can be implemented as to run in time* O(|E|).

We also compare our algorithm with previous versions of EADAM through computational experiments. In Section 7.4, the theoretical advantage of student-rotate-remove is verified computationally on random instances.

#### 7.1. Kesten's EADAM

**Algorithm 7.1** (Kesten's EADAM)

**Require:**  $(G(A \cup B, E), <, \mathbf{q})$ , consenting students  $\overline{A} \subseteq A$ 

- 1: Let  $G^0 = G$ , i = 0.
- 2: Run student-proposing Gale–Shapley's algorithm on  $(G^i, <, \mathbf{q})$  to obtain assignment  $M^i$ .
- 3: **while** there is a consenting interrupter, **do**
- 4: Identify the maximum k' such that there exists a consenting interrupter at step k'.
- 5: Let *E*<sup>'</sup> be the set of all interrupting pairs *ab* at step *k*<sup>'</sup> such that *a* is consenting.
- 6: Define  $G^{i+1}$  from  $G^i$  by removing edges in E'. Set i = i + 1.
- 7: Run student-proposing Gale–Shapley's algorithm on  $(G^i, <, \mathbf{q})$  to obtain assignment  $M^i$ .
- 8: end while
- 9: Output  $M^i$ .

Recall that Gale–Shapley's algorithm (with students proposing) is executed in successive steps. During each step, every student who is currently unmatched applies to the first school in the sutdent's preference list to which the student has not yet applie, and gets either temporarily accepted or rejected. A student a is called an *interrupter* (for school b at step k') if a is temporarily accepted by school b at some step k < k', a is rejected by school b at step k', and there exists a student that is rejected by school b during steps  $k, k+1, \dots, k'-1$ . In such case, we also call ab an interrupting pair (at step k'). Informally speaking, an interrupter is a student who, by applying to school b, interrupts a desirable assignment between school b and another student at no gain to the student's self. Removing such interruptions is crucial in neutralizing their adverse effects on the outcome. Demonstration of these concepts can be found in Example 7.1.

Kesten's EADAM takes as input an instance  $(G, <, \mathbf{q})$  with  $G = (A \cup B, E)$  and a set  $\overline{A} \subseteq A$  of students that

we call *consenting*. Each iteration of EADAM starts by running Gale–Shapley's algorithm from scratch. It then removes from the graph certain interrupting pairs involving consenting interrupters. The algorithm terminates when there are no interrupting pairs whose corresponding interrupters are consenting students.

Details of Kesten's algorithm can be found in Algorithm 7.1 and an illustration of the algorithm can be found later in Example 7.1.

The following theorem collects some results from Kesten (2010) and Tang and Yu (2014), demonstrating the transparency of the consenting incentives and some attractive properties of EADAM's output. Recall that an assignment M is constrained efficient if it does not violate any nonconsenting students' priorities, but any other assignment M' that dominated M does.

#### Theorem 7.2. Under Kesten's EADAM,

- 1. The assignment of a student does not change whether the student consents or not. That is, for any student  $a \in A$  and any set of consenting students  $\overline{A} \subseteq A$ , if M and M' are the outputs of EADAM on inputs  $\{(G, <, \mathbf{q}), \overline{A} \setminus \{a\}\}$  and  $\{(G, <, \mathbf{q}), \overline{A}\}$ , respectively, then M(a) = M'(a).
- 2. The output is Pareto efficient when all students consent and is constrained efficient otherwise.

**Example 7.1.** Each school in this example has a quota of one. The preference lists are given as follows. All students are consenting except for  $a_3$ .

$$a_1$$
:  $b_1 > b_2 > b_3 > b_4$   $b_1$ :  $a_4 > a_2 > a_1 > a_3$   
 $a_2$ :  $b_1 > b_2 > b_3 > b_4$   $b_2$ :  $a_2 > a_3 > a_1 > a_4$   
 $a_3$ :  $b_3 > b_2 > b_4 > b_1$   $b_3$ :  $a_1 > a_4 > a_3 > a_2$   
 $a_4$ :  $b_3 > b_1 > b_2 > b_4$   $b_4$ :  $a_3 > a_1 > a_2 > a_4$ 

**Gale–Shapley's Algorithm.** The student-proposing Gale–Shapley's algorithm outputs the assignment  $M^0 = \{a_1b_3, a_2b_2, a_3b_4, a_4b_1\}$ . Steps of the algorithm are given as follows:

step	$b_1$	$b_2$	$b_3$	$b_4$
1	Ø(, a₂		<b>⋈</b> , a₄	
2		$\langle \chi \rangle_1, a_3$		
3			$a_1, \cancel{a_4}$	
4	$a_2, a_4$			
5		$a_2, \lambda_3$		
6				$a_3$

**Iteration 1.** From the steps of Gale–Shapley's algorithm, one can identify all interrupting pairs. For instance,  $a_2$  proposes to  $b_1$  at step 1. This causes  $a_1$  to be rejected by  $b_1$ . However,  $a_2$  is later rejected by  $b_1$  at step 4. Thus, by definition,  $a_2b_1$  is an interrupting pair at step 4.

In total, there are three interrupting pairs,  $a_3b_2$ ,  $a_2b_1$ ,  $a_4b_3$  from the last step to the first. The last interrupting pair of a consenting interrupter is  $a_2b_1$  given that  $a_3$  is not a consenting student. Thus, k' = 4. Because there is only one interrupting pair at step k' = 4, EADAM simply removes  $a_2b_1$  from the instance. On the new instance, EADAM reruns Gale–Shapley's algorithm, and the resulting assignment is  $M^1 = \{a_1b_1, a_2b_2, a_3b_4, a_4b_3\}$ .

**Iteration 2.** One can check that there are no interrupting pairs and, thus, no consenting interrupters. Hence, EA-DAM terminates and outputs assignment  $M^1$ .

Note that, using tools developed in previous sections, one can show that  $a_2b_1$ , the first edge that is removed by EADAM, is actually a legal edge, and the assignment output of EADAM,  $M^1$ , is not a legal assignment.  $\diamondsuit$ 

#### 7.2. School-Rotate-Remove with Consent

**Algorithm 7.2.** (School-Rotate-Remove with Consent) **Require:**  $(G(A \cup B, E), <, \mathbf{q})$ , consenting student

**Require:**  $(G(A \cup B, E), <, \mathbf{q})$ , consenting students  $\overline{A} \subseteq A$ 

- 1: Find the student-optimal stable assignment  $M_0$  of  $(G, <, \mathbf{q})$  via Gale–Shapley's algorithm.
- 2: Let  $G^0 := G$  and  $M^0 := M_0$ .
- 3: Set i = 0 and let  $D^0$  to be the school-rotation digraph of  $M^0$  in  $(G^0, <, \mathbf{q})$ .
- 4: **while**  $D^i$  still has an arc, **do**
- 5: Find (i) arcs (b', a) and  $(a, b) \in A(D^i)$ , where b is a sink in  $D^i$  or (ii) a cycle  $C^i$  of  $D^i$ .
- 6: **if** (i) is found, **then**
- 7: Define  $G^{i+1}$  from  $G^i$  by removing ab' and set  $M^{i+1} = M^i$ .
- 8: **if**  $a \notin \overline{A}$ , **then**
- 9: Remove from  $G^{i+1}$  edges a'b' for all a' such that  $a>_{h'}a'$ .
- 10: **end if**
- 11: **else if** (ii) is found, **then**
- 12: Let  $\rho^i$  be the corresponding school rotation. Set  $M^{i+1} = M^i/\rho^i$  and  $G^{i+1} = G^i$ .
- 13: **end if**
- 14: Set i = i + 1 and let  $D^i$  be the school-rotation digraph of  $M^i$  in  $(G^i, <, \mathbf{q})$ .
- 15: end while
- 16: Output *M*<sup>1</sup>.

Morrill (2016) shows that, when all students consent, the output of EADAM is the student-optimal legal assignment. Hence, school-rotate-remove can be employed to find this assignment in time O(|E|) (see Theorem 5.2). However, as Example 7.1 shows, when only some students consent, EADAM may output an assignment that is not legal. We show in this section how to suitably modify school-rotate-remove in order to obtain the assignment output of EADAM for

any given set of consenting students without sacrificing the running time.

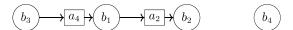
In school-rotate-remove, the key idea is to reroute arcs that point to students who are assigned to sinks in the rotation digraph. This allows us to identify school rotations in the underlying legalized instance  $(G_L, <, \mathbf{q})$ . Assume, for example, that  $(b',a),(a,b) \in A(D_B)$ , and b is a sink. Upon such rerouting, a's priority might be violated. In particular, if b' successfully participates in a school rotation after the rerouting, then ab' is a blocking pair for the new assignment. Hence, under the EADAM framework, if a is not consenting, we can no longer freely reroute arcs pointing to a. In fact, in order to respect a's priority (i.e., to avoid ab' becoming a blocking pair), b' cannot be assigned to any student a' such that  $a >_{b'} a$ '. This means that the arc coming out of b' cannot be rerouted to any other student, essentially making b' a sink.

A detailed description of our algorithm is presented in Algorithm 7.2. Throughout the rest of the section, we call school rotations simply rotations. As in Algorithm 5.1, when both cases (i) and (ii) are present at step 5 of some iteration, we are free to choose between steps 6 and 11. These choices do not affect the final assignment output. We formalize this statement in Theorem EC.7.1. A step-by-step application of our algorithm on the instance from Example 7.1 is outlined in Example 7.2.

**Example 7.2.** Consider the instance given in Example 7.1. From the student-optimal stable assignment  $M^0 := \{a_1b_3, a_2b_2, a_3b_4, a_4b_1\}$ , we can construct the rotation digraph as follows. Note that, in this graph and in the following, some isolated nodes are not included.

$$(b_3)$$
  $a_4$   $b_1$   $a_2$   $b_2$   $a_3$   $b_4$ 

**Iteration 1.** Because  $b_4$  is a sink, we remove edge  $a_3b_2$  as in step 7 in the hope of rerouting the arc coming out of  $b_2$ . However, because  $a_3$  is not consenting, we have to additionally remove edges  $a_1b_2$  and  $a_4b_2$  as in step 9. This completely removes the possibilities of rerouting, essentially making  $b_2$  a sink as seen in the rotation digraph of the updated instance.



**Iteration 2.** Now,  $b_2$  is a sink. Because its assigned student  $a_2$  is consenting, the algorithm simply removes edge  $a_2b_1$  in step 7, resulting in the following updated rotation digraph.



**Iteration 3.** We can now eliminate the rotation (i.e., trading schools between  $a_1$  and  $a_4$ ) and update the assignment to be  $\{a_1b_1, a_2b_2, a_3b_4, a_4b_3\}$ . After the assignment update, the new rotation digraph only contains sinks, and thus, the algorithm terminates.

This final assignment coincides with the assignment output of EADAM. ♦

In our rotation-based algorithm, the students from whom we seek consent are those who are assigned to schools corresponding to sinks, and thus, they are not in any directed cycles in the current and subsequent rotation digraphs. Therefore, there is a clear separation between the students from whom we ask for consent and those participating in Pareto-improvement cycles (i.e., school rotations). This is consistent with the result in Theorem 7.2, part 1, that students have no incentive to not consent.

The proof of the following statement can be found in E-companion EC.7.

**Theorem 7.3.** For any given input, the outputs of Algorithms 7.2 and 7.1 coincide.

We remark that the proof of Theorem 7.3 is different (and quite harder) than the proof of Theorem 5.1. Indeed, for the latter, we can build on the fact that legal assignments form a lattice although in the former, we do not have such a well-behaved structural result at our disposal. Hence, a careful analysis of the algorithms is needed.

# 7.3. Fast Implementation of School-Rotate-Remove with Consent

The fast implementation is a modification of that of Algorithm 5.1. Therefore, we defer the proof of Lemma 7.1 to E-companion EC.6.4. An example demonstrating the implementation can also be found in E-companion EC.6.3.

**Lemma 7.1.** Algorithm 7.2 can be implemented so as to run in time O(|E|).

**Proof of Theorem 7.1.** This follows immediately from Theorem 7.3 and Lemma 7.1.  $\Box$ 

#### 7.4. Computational Experiments

Because Gale–Shapley's algorithm on stable assignment instances can be implemented to run in time O(|E|) (see Gusfield and Irving 1989, Manlove 2013), the original EADAM (Kesten 2010) runs in time  $O(|E|^2)$  because it runs Gale–Shapley's routine at most |E| times. A simplified version of EADAM (Tang and Yu 2014), for which the details are presented in the e-companion, runs in time O(|E||V|) because it runs Gale–Shapley's routine at most |V| times. We remark that, although mechanism design rather than computational complexity, is the

primary interest of Kesten's (2010) paper, computational efficiency is nevertheless crucial in putting the mechanism into practice, especially for large markets, such as the New York school system. In fact, Tang and Yu (2014) mention computational tractability as one of their contributions.

One major advantage of our school-rotate-remove with consent is that, instead of repeatedly running Gale–Shapley's algorithm, we update the assignment locally using the structural results (lattice structure and rotations) of stable assignments. Our algorithm runs in time O(|E|) as shown in Lemma 7.1.

To further demonstrate the computational advantage of our algorithm, we randomly generated instances of varying sizes and recorded the running time of all three algorithms. The running time of Gale-Shapley's algorithm is also recorded as a benchmark. The number of students in our instances ranges from 500 to 30,000, and the corresponding number of schools ranges from 5 to 300. For each instance size, 100 instances  $(G, <, \mathbf{q})$  are obtained by randomly generating < and  $\mathbf{q}$ . For each student a, the preference list  $<_a$  is defined by a random permutation of B. The preference lists of schools are similarly defined. The quota of each school is randomly selected between 50 and 150 uniformly. Note that, in this set of simulations, students and schools have complete preference ranking of the opposite side. That is, in all instances, G is a complete bipartite graph. We also conduct another set of simulations (details later) with incomplete preference lists. We tested scenarios in which each student is randomly determined to be consenting with probability 10%, 30%, 50%, 80%, and 100%. The experiments were carried out on a computing node with one core and 4 GB RAM.

A visual representation of the running times of different algorithms can be found in Figure 2. The shaded areas are 95% confidence intervals of each algorithm for given instance sizes. Our algorithm performs significantly faster than the simplified EADAM (Tang and Yu 2014) and dramatically faster than the original EADAM (Kesten 2010) with the differences being especially pronounced when all students consent.

The New York City school district has approximately 90,000 students applying to 700 public high school programs every year, and students can list up to 12 schools in their application (Narita 2016). We further conducted computational experiments whose instance sizes are similar to those of New York City. We compared our algorithm with simplified EADAM on random instances generated similarly as previously described. However, in this set of simulations, we fix instance size with |A| = 90,000 and |B| = 700.

Moreover, the quota of each school is selected uniformly at random from integers between  $[0.5 \times \mu]$  and  $[1.5 \times \mu]$ , where  $\mu = \lceil \frac{|A|}{|B|} \rceil$ . In generating < , for every student a,  $<_a$  is obtained by truncating the random permutation such that only the top 12 schools are listed; for every school b,  $<_b$  is obtained by restricting the random permutation to students who have b in their preference lists. Graph G can be deduced from the preference lists. Results of our experiments are summarized in Figure 1. The difference in computational time is noticeably different from all levels of consenting percentages. In particular, when all students consent, school-rotate-remove takes approximately three minutes, whereas simplified EADAM takes, on average, four hours, and its run time has a much higher variance.

## 8. Legal Matchings and Latin Marriages

In this section, we restrict to one-to-one instances. For an instance (G, <) of the stable marriage problem, let S(G, <) and  $\mathcal{L}(G, <)$  denote the set of stable and legal matchings, respectively. In addition, we call  $(G_L, <)$  the *legalized instance* of (G, <), where  $G_L$  is the subgraph of G defined as in Theorem 3.1. We say an instance (G, <) is legal if  $G_L = G$ .

An  $n \times n$  matrix is a *Latin square* if each row and column are a permutation of numbers 1,2, ..., n. Given an instance (G, <) of the stable marriage problem with complete lists, we call the position of a in the preference list of b the rank of a in b's list. Following the work of Benjamin et al. (1995), we say an instance (G, <) with |A| = |B| = n is *Latin* if there exists a Latin square Q with n rows indexed by elements of A and A columns indexed by elements of A such that, for each row A and column A0, A1 is the rank of A2 in A3 list, and A4 and A5 list. We call such A6 the *Latin ranking matrix*6. See Example 8.1 for an example of a Latin ranking matrix and its associated stable marriage instance. In this section, we prove the following.

**Theorem 8.1.** Let (G, <) be a Latin instance. Then,  $G_L = G$  and there exists an instance (G', <') with an additional man  $\widetilde{a}$  and an additional woman  $\widetilde{b}$  such that |S(G', <')| = 1 and  $\mathcal{L}(G', <') = \{M \cup \{\widetilde{ab}\} : M \in S(G, <)\}$ .

Benjamin et al. (1995) construct, for each even n, a Latin instance (G, <) with n men and n women such that  $|S(G, <)| = \omega(2^n)$ , and in the man-optimal stable matching, each man is given his favorite partner. Hence, Theorem 8.1 implies that, for each odd n, there is an instance (G', <') with n men and n women such that |S(G', <')| = 1 and  $|L(G', <')| = \omega(2^n)$ —that is, it has one stable matching but exponentially many legal

matchings. Moreover, proofs of our construction for (G', <') shows that the man-optimal legal matching in  $\mathcal{L}(G', <')$  assigns to each man from G his favorite partner, and the stable matching in  $\mathcal{S}(G', <')$  assigns to each man from G his second least favorite partner (see Lemma 8.2). Note that, up to a different constant in the basis, the asymptotic ratio between the quantities  $|\mathcal{L}(G, <)|$  and  $|\mathcal{S}(G, <)|$  cannot be increased as it has been recently shown that there exists an absolute constant c > 1 such that every instance of the stable marriage problem with n men and n women has  $O(c^n)$  stable matchings (Karlin et al. 2018).

We believe that future investigations of the relationship between Latin instances and legal matchings may provide further advancement on a question by Knuth (1976). In his seminal work, Knuth (1976) asks for a characterization of instances that maximize |S(G, <)| for each value of  $|A| = |B| = n \in \mathbb{N}$ . Although an asymptotic upper bound follows from the work cited here (Karlin et al. 2018), the characterization of these instances is unsolved even for reasonable small sizes. Note that, for each  $n \in \mathbb{N}$ , there is always a legal instance achieving the maximum, and as for any instance (G, <), we have  $|S(G_L, <)| = |\mathcal{L}(G_L, <)| \ge |S(G, <)|$ .

The following theorem (Benjamin et al. 1995) gives a necessary and sufficient condition for a matching to be stable in a Latin instance.

**Theorem 8.2.** Let M be a matching of the instance defined by a Latin ranking matrix Q. M is stable if and only if there do not exist row a and column b such that Q(M(b),b) > Q(a,b) > Q(a,M(a)) or Q(M(b),b) < Q(a,b) < Q(a,M(a)).

The following lemma shows that every Latin instance is legal.

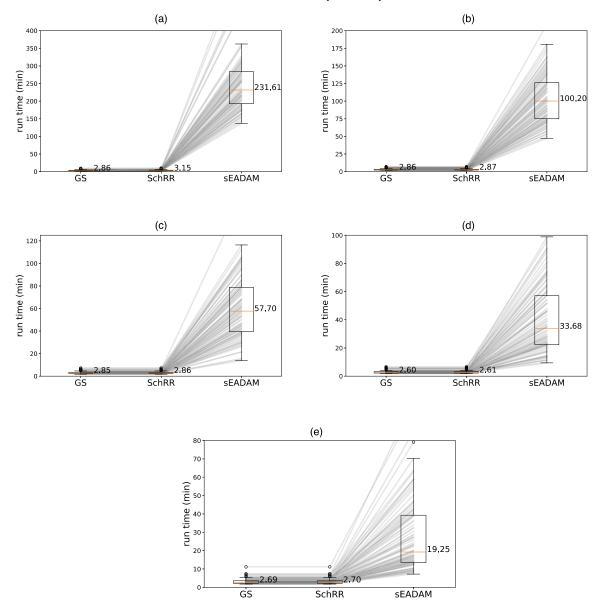
**Lemma 8.1.** Let (G, <) be a Latin instance. Then,  $G_L = G$ .

**Proof.** Assume Q is the Latin ranking matrix of instance (G, <) and  $Q \in \mathbb{Z}^{n \times n}$ . For  $i \in [n]$ , let  $M^i = \{ab : Q(a,b) = i\}$ . By definition of Latin squares,  $M^i$  is a matching. By construction, for any row a and column b,  $Q(M^i(b),b) = i = Q(a,M^i(a))$ . Therefore,  $M^i$  must be stable and, thus, legal because of Theorem 8.2. Because  $\bigcup_{i \in [n]} M^i = E(G)$ , by Theorem 3.1,  $G_L = G$ .  $\square$ 

As we show next, the set of stable matchings of a Latin instance can be "masked" into the set of legal matchings of an auxiliary instance with only one more man and one more woman such that the auxiliary instance has only one stable matching. The construction is as follows: given a Latin instance  $(G(A \cup B, E), <)$ , construct an auxiliary instance  $(G'(A' \cup B', E'), <')$ , where  $A' = A \cup \{\overline{a}\}, B' = B \cup \{\overline{b}\}, E' = A' \times B'$ , and <' is defined as follows:

i. Every  $a \in A$  ranks b in the last position, and  $<'_a$  restricted to B is exactly  $<_a$ .

**Figure 1.** (Color online) Comparing Simplified EADAM (sEADAM) and School-Rotate-Remove with Consent (SchRR) on Random Instances Whose Sizes Are Similar to Those of the New York City School System



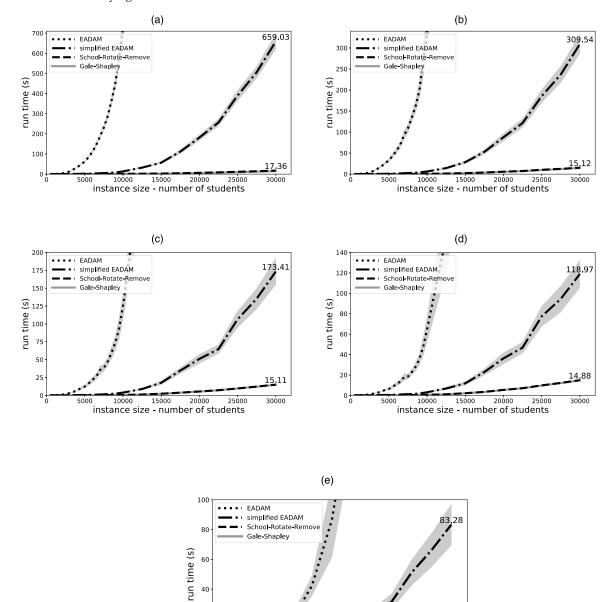
Notes. Run times of Gale–Shapley's algorithm (GS) are included as a benchmark. Run time of Kesten's original algorithm is not included because most instances fail to finish within 24 hours. Each line represents one instance. Box plots and averages of run times are included for each algorithm. (a) All students consent. (b) Eighty percent of students consent. (c) Fifty percent of students consent (d) Thirty percent of students consent. (e) Ten percent of students consent.

- ii.  $\tilde{a}$  has an arbitrary ranking of B' as long as b is the least preferred.
- iii. Every  $b \in B$  ranks  $\tilde{a}$  in the second place, and  $<'_b$  restricted to A is exactly  $<_b$ .
- iv.  $\widetilde{b}$  has an arbitrary ranking of A' as long as  $\widetilde{a}$  is ranked first.

An example of our construction can be found in Example 8.1.

**Example 8.1.** Consider the following Latin ranking matrix Q and the associated instance (G, <).

Figure 2. Comparing EADAM, Simplified EADAM, and School-Rotate-Remove with Consent in the One-to-Many Setting on Random Instances of Varying Sizes



Notes. Average run time of simplified EADAM and school-rotate-remove with consent included for the largest instance in our experiment. (a) All students consent. (b) Eighty percent of students consent. (c) Fifty percent of students consent. (d) Thirty percent of students consent. (e) Ten percent of students consent.

10000 15000 20000 25000 instance size - number of students

25000

30000

$a_1$ : $b_1 > b_2 > b_3 > b_4$	$b_1$ : $a_4 > a_3 > a_2 > a_1$
$a_2$ : $b_2 > b_1 > b_4 > b_3$	$b_2$ : $a_3 > a_4 > a_1 > a_2$
$a_3$ : $b_3 > b_4 > b_1 > b_2$	$b_3$ : $a_2 > a_1 > a_4 > a_3$
$a_4$ : $b_4 > b_3 > b_2 > b_1$	$b_4$ : $a_1 > a_2 > a_3 > a_4$

20

Consider the matching  $M = \{a_1b_1, a_2b_3, a_3b_2, a_4b_4\}$ , which corresponds to the boxed cells in the Latin ranking matrix. M is not stable because, as one can check,  $a_3b_1$  is a blocking pair. Equivalently, we can apply Theorem 8.2 on the Latin ranking matrix with  $a = a_3, b =$  $b_1$  and conclude that M is not stable. In particular, we have Q(M(b), b) = 1 < Q(a, b) = 3 < Q(a, M(a)) = 4.

One can check that (G, <) has 10 stable matchings. Now consider the auxiliary instance (G', <'). Note that its preference lists are exactly those given in Example EC.6.1 with  $a_5 = \tilde{a}$  and  $b_5 = \tilde{b}$ . (G', <').

The auxiliary instance has only one stable matching, which is  $\{a_1b_4, a_2b_3, a_3b_2, a_4b_1, \widetilde{ab}\}$ , but its legalized instance  $(G'_1, <'_1)$  has 10 stable matchings.  $\diamondsuit$ 

Before concluding the proof of Theorem 8.1, we first show the following facts.

**Lemma 8.2.** Given a Latin instance (G, <) with  $G = (A \cup B, E)$ , define (G', <') as before. Then, |S(G', <')| = 1, and each man from A is given his second least favorite partner (with respect to <') in the unique stable matching of (G', <').

**Proof.** Let  $M \in \mathcal{S}(G', <')$  be a stable matching in the auxiliary instance. We first show  $M(b) = \tilde{a}$ . Assume by contradiction that M(b) = a for some  $a \in A$ . Let b be a's least preferred partner in B. Then,  $b >_a' b = M(a)$  by construction. By the symmetric nature of Latin instances, a must be b's most preferred partner in A, which means  $a >_{b}' M(b)$ . But, then, ab is a blocking pair of M, contradicting stability. Next, we want to show every woman in B is matched to her most preferred man. Assume by contradiction that the claim is not true for some  $b \in B$ . Then,  $\widetilde{a} >_b' M(b)$ . Because  $b > \stackrel{\cdot}{\sim} b$  by construction,  $\widetilde{a}b$  blocks M, which again contradicts stability. Hence, S(G', <') contains exactly one stable matching, namely, the one in which every woman is matched to her most preferred man according to <'. That is, every man  $a \in A$  is given his second least favorite partner with respect to <'.  $\square$ 

**Lemma 8.3.** Let (G, <) and (G', <') be as before with  $G = (A \cup B, E)$  and  $G' = (A' \cup B', E')$ . Then,  $\mathcal{L}(G', <') = \{M \cup \{\widetilde{ab}\} : M \in \mathcal{S}(G, <)\}$ .

**Proof.** Let  $M_0$  be the only stable matching of (G', <'). Because every woman in B' is matched to her most preferred man in A' as shown in the proof of Lemma 8.2,  $M_0$  is also the woman-optimal legal matching of  $\mathcal{L}(G', <')$ . In addition, because b is the least preferred woman of every man by construction of G', b is a sink in the woman-rotation digraph of  $M_0$  and remains a sink throughout the execution of woman-rotateremove. Thus,  $\tilde{a}$  is matched to b in the man-optimal legal matching of  $\mathcal{L}(G', <')$ . Hence,  $\tilde{a}b \in M$  for all  $M \in$  $\mathcal{L}(G'_{\perp}, <')$  and according to Theorem 3.1, all edges in E: =  $\{ab : a \in A\} \cup \{\tilde{a}b : b \in B\}$  (i.e., edges that are adjacent to exactly one of  $\tilde{a}$  and b) are illegal. By Lemma 3.1, we have  $\mathcal{L}(G', <') = \mathcal{L} (G'[E' \setminus E], <') = \{M \cup \{\tilde{a}b\}: \}$  $M \in \mathcal{L}(G, <)$ , where the last equality is because  $E \setminus E = E(G) \cup \{\tilde{a}b\}$ . Finally, by Lemma 8.1, we have  $\mathcal{L}(G, <) = \mathcal{S}(G, <)$  and, thus,  $\mathcal{L}(G', <') = \{M \cup \{\tilde{a}b\}:$  $M \in \mathcal{S}(G, <)$ .

**Proof of Theorem 8.1.** This is immediately implied by Lemmas 8.1–8.3.  $\square$ 

#### Acknowledgments

The authors thank Yiannis Mourtos and Jay Sethuraman for useful suggestions on an earlier version of this draft as well as three anonymous referees for pointing to the relevant literature and comments that improved the organization and broadened the scope of the manuscript.

#### **Endnotes**

- <sup>1</sup> Priorities are preferences with ties as schools usually rank students based on categorical information, such as demographics, test scores, etc.
- <sup>2</sup> In some literature, Gale–Shapley's algorithm is referred to as *deferred acceptance*. In this paper, we stick to Gale–Shapley.
- <sup>3</sup> Indeed, even though Birkhoff's (1937) representation theorem implies that there is a bijection between the elements of a distributive lattice and the closed sets of an associated poset, it is not clear how to use this information algorithmically. A typical example are strongly stable matchings, which have been known for a long time to form a distributive lattice (Manlove 2002), but only recently was this structure exploited for algorithmic purposes (Kunysz et al. 2016). See Faenza and Zhang (2020) for sufficient conditions on algorithmic exploitation of Birkhoff's (1937) theorem.
- <sup>4</sup> (1) See https://github.com/xz2569/LegalAssignments. (2) See https://github.com/xz2569/FastEADAM.
- <sup>5</sup> This notion of an assignment blocking another assignment is not standard and is adopted from Morrill (2016).
- <sup>6</sup> It is worth noticing that the definition of both student and school rotations can be simplified but in different ways. However, in order to keep the treatment compact, we give a unique presentation encompassing both.
- <sup>7</sup> If  $D_B$  has both a sink and a directed cycle, the algorithm is free to choose between the two cases.
- <sup>8</sup> A student a's priority is violated at assignment M if there is a school b such that ab is a blocking pair of M.

# References

Abdulkadiroğlu A, Sönmez T (2003) School choice: A mechanism design approach. *Amer. Econom. Rev.* 93(3):729–747.

Abdulkadiroğlu A, Che YK, Yasuda Y (2015) Expanding "choice" in school choice. *Amer. Econom. J. Microeconomics* 7(1):1–42.

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.

Afacan MO, Aliogullari ZH, Barlo M (2017) Sticky matching in school choice. Econom. Theory 64(3):509–538.

Ashlagi I, Nikzad A (2020) What matters in school choice tie-breaking? How competition guides design. J. Econom. Theory 190:105120.

Baïou M, Balinski M (2004) Student admissions and faculty recruitment. Theoretical Comput. Sci. 322(2):245–265.

Bando K (2014) On the existence of a strictly strong Nash equilibrium under the student-optimal deferred acceptance algorithm. *Games Econom. Behav.* 87:269–287.

Bansal V, Agrawal A, Malhotra VS (2007) Polynomial time algorithm for an optimal stable assignment with multiple partners. *Theoretical Comput. Sci.* 379(3):317–328.

Benjamin AT, Converse C, Krieger HA (1995) How do I marry thee? Let me count the ways. *Discrete Appl. Math.* 59(3): 285–292.

Birkhoff G (1937) Rings of sets. Duke Math. J. 3(3):443-454.

Dur U, Gitmez AA Yilmaz O (2019) School choice under partial fairness. *Theoret. Econom.* 14(4):1309–1346.

Ehlers L (2007) von Neumann–Morgenstern stable sets in matching problems. *J. Econom. Theory* 134(1):537–547.

Ehlers L, Morrill T (2020) (II)legal assignments in school choice. *Rev. Econom. Stud.* 87(4):1837–1875.

- Erdil A, Ergin H (2008) What's the matter with tie-breaking? Improving efficiency in school choice. Amer. Econom. Rev. 98(3): 669–689.
- Ergin HI (2002) Efficient resource allocation on the basis of priorities. *Econometrica* 70(6):2489–2497.
- Faenza Y, Zhang X (2020) Affinely representable lattices, stable matchings, and choice functions. Singh M, Williamson D, eds. Proc. IPCO 2021 (Springer, Cham), 89–103.
- Gale D, Shapley LS (1962) College admissions and the stability of marriage. Amer. Math. Monthly 69(1):9–15.
- Gusfield D (1987) Three fast algorithms for four problems in stable marriage. SIAM J. Comput. 16(1):111–128.
- Gusfield D, Irving RW (1989) The Stable Marriage Problem: Structure and Algorithms (MIT Press, Cambridge, MA).
- Irving RW, Leather P, Gusfield D (1987) An efficient algorithm for the optimal stable marriage. *J. ACM* 34(3):532–543.
- Karlin AR, Gharan SO, Weber R (2018) A simply exponential upper bound on the maximum number of stable matchings. Henzinger M, ed. Proc. 50th Annual ACM SIGACT Sympos. Theory Comput. (ACM, New York), 920–925.
- Kesten O (2010) School choice with consent. Quart. J. Econom. 125(3): 1297–1348.
- Kloosterman A, Troyan P (2016) Efficient and Essentially Stable Assignments (University of Virginia).
- Knuth DE (1976) Mariages Stables et Leurs Relations avec d'autres Problèmes Combinatoires: Introduction à l'analyse Mathématique des Algorithmes (Presses de l'Université de Montréal, Montréal).
- Kunysz A, Paluch K, Ghosal P (2016) Characterisation of strongly stable matchings. Krauthgamer R, ed. Proc. 27th Annual ACM-SIAM Sympos. Discrete Algorithms (Society for Industrial and Applied Mathematics, Philadelphia), 107–119.
- Manlove DF (2002) The structure of stable marriage with indifference. *Discrete Appl. Math.* 122(1–3):167–181.
- Manlove DF (2013) Algorithmics of Matching Under Preferences, vol. 2 (World Scientific).
- Morrill T (2016) Which school assignments are legal? Technical report, University of North Carolina.

- Narita Y (2016) Match or mismatch: Learning and inertia in school choice. Working paper, Yale University.
- Roth AE (1984) The evolution of the labor market for medical interns and residents: A case study in game theory. *J. Political Econom.* 92(6):991–1016.
- Roth AE, Peranson E (1999) The redesign of the matching market for American physicians: Some engineering aspects of economic design. *Amer. Econom. Rev.* 89(4):748–780.
- Roth AE, Rothblum UG (1999) Truncation strategies in matching markets—In search of advice for participants. *Econometrica* 67(1):21–43.
- Roth AE, Sotomayor M (1992) Two-sided matching. *Handbook of Game Theory with Economic Applications*, vol. 1, (Amsterdam), 485–541.
- Roth AE, Sönmez T, Ünver MU (2005) Pairwise kidney exchange. *J. Econom. Theory* 125(2):151–188.
- Tang Q, Yu J (2014) A new perspective on Kesten's school choice with consent idea. J. Econom. Theory 154:543–561.
- Tang Q, Zhang Y (2021) Weak stability and Pareto efficiency in school choice. Econom. Theory 71(2):533–552.
- Von Neumann J, Morgenstern O (1953) *Theory of Games and Economic Behavior*. 3rd ed. (Princeton University Press, Princeton, NJ).
- Wako J (2010) A polynomial-time algorithm to find von Neumann-Morgenstern stable matchings in marriage games. *Algorithmica* 58(1):188–220.

**Yuri Faenza** is an associate professor in the department of industrial engineering and operations research at Columbia University. His research focuses on discrete optimization and operations research, in which he works on both fundamental properties and efficient algorithms.

**Xuan Zhang** is currently a postdoctoral researcher at Facebook. She completed her doctoral thesis in the department of industrial engineering and operations research at Columbia University. She is broadly interested in algorithms and mechanism designs for networked marketplaces and, more generally, in the interplay between economics, optimization, and computer science.