Submitted to
manuscript 0001

Efficient and (pretty) fair course assignment with quotas

Martin Bichler, Alexander Hammerl, Stefan Waldherr
Department of Informatics, Technical University of Munich, Germany, {bichler, hammerl, waldherr}@in.tum.de

Thayer Morill
Department of Economics, NC State University, USA, thayermorrill@gmail.com

We consider the problem of assigning students to courses which is arguably one of the most common instances of object assignment without money. It is well-known that it is impossible to combine the three properties strategyproofness, efficiency and fairness. In other applications, fairness (or envy-freeness) is emphasized over efficiency; however, for large-scale course assignment applications efficiency appears to be the primary consideration. A second feature of most course assignment is that courses must be allocated a minimum number of students. We introduce modifications of the Top Trading Cycles algorithm which significantly reduce the instances of justified envy while accommodating minimum capacities. Our improvements are based on the following two observations: TTC myopically chooses which students will trade and students may trade even after they are guaranteed their top choice. We leverage field data from a large-scale course assignment application and show that our algorithm significantly reduces the amount of justified envy while still maintaining efficiency and strategyproofness for students.

Key words: Top Trading Cycles, Course Assignment, Assignment

Version December 13, 2018

1. Introduction

Matching with preferences has developed into a rich theoretical literature with many real-world applications ranging from the allocation of medical residents to hospitals to course assignment and school choice (Roth 2008, 2002, Manlove 2013). Efficiency, fairness, and strategyproofness are seen as the key design desiderata. Unfortunately, it is well-known that efficiency and fairness are incompatible in the sense that there may not always exist an assignment that is both fair and efficient (Roth 1982a).

A market designer typically faces a decision between Gale's Top Trading Cycles algorithm (hereafter TTC), which is strategyproof and efficient but not fair, and Gale and Shapley's Deferred Acceptance algorithm (hereafter DA), which is strategyproof and fair but is not efficient. Abdulkadiroğlu and Sönmez (2003) discussed in the efficiency/fairness tradeoff their seminal paper on school assignment. Their recommendation regarding DA is clear; DA Pareto dominates any other fair assignment. However, their recommendation regarding TTC is far less certain. Surprisingly little is known about TTC or efficient allocations and their fairness when multiple units of an objects are assigned to more than one agent, e.g. many students are assigned to the same course. A natural question is how can we make an efficient mechanism as fair as possible.

We will demonstrate that we can assign agents to objects in a significantly fairer way than TTC. This is our first technical contribution. Our second technical contribution is to demonstrate how to incorporate minimum quotas in this assignment problem, which is typically a requirement in course allocation. Our practical contribution is to measure the impact of these changes using data from a large, real-world course assignment problem. We find that the mechanism we suggest substantially improves the fairness of the assignment compared to TTC and quota-respecting versions of TTC. The results led to a policy change in a large-scale course assignment application at the Technical University of Munich (TUM), which we discuss next.

1.1. Requirements in Course Assignment

Course assignment is probably one of the most wide-spread applications for matching with preferences as it occurs regularly in almost all educational institutions. At TUM, course assignments are handled within a department. Our assignment was for the Informatics department at TUM (IN.TUM) which is currently the largest department with more than 6,000 students. Students have to enroll in seminars and practical courses, and the assignment has grown to a total number of 2,000-3,000 students per semester. In addition, the system is used by large classes with several hundred students for the assignment of students to tutor groups. These course-specific assignments add up to another 7,000 matchings of students to courses every year only in the Informatics department. Based on its success, the matching system was also adopted by other departments such as mathematics and mechanical engineering.

Previously, students had been assigned based on a first-come-first-served (FCFS) basis, but due to severe shortcomings of the FCFS system, the assignment mechanism was changed to DA in 2014 (Diebold and Bichler 2017). In FCFS, students would often sign up for dozens of courses, but then cancel the registrations for all but their most preferred among those courses where she was admitted. The department of Informatics was looking for a more principled approach to course assignment and implemented DA. Organizing the course assignment via DA was considered a significant improvement by the student union and the faculty of the department and overall a great success. However, DA does not allow for *minimum quotas*, which is important in most course assignment problems, and it is *not efficient*. These deficiencies recently motivated the department to rethink the assignment mechanism.

First, efficiency is seen as a first-order concern in large-scale course assignment applications by the university administration. It is rarely the case in large-scale applications that a student would have sufficient information about the priorities of other students to even know if she has justified envy. On the other hand, efficient use of scarce resources is a key concern to the dean of studies. Moreover, the course assignment environment is very different from more common assignment problems in school choice where preferences and priorities are often homogenous. The students have diverse preferences over courses, mirroring their individual interests in all areas of computer science and the breadth of available topics. Likewise, courses have diverse priorities over students (e.g. their type of study) and lecturers rank individual students based on these priorities. In school assignment applications where the preferences and priorities are largely homogeneous, the DA assignment is very close to Pareto efficient (see for example (Abdulkadiroğlu et al. 2017) which describes such assignments in both Boston and New Orleans). However, as we will show, DA is quite inefficient in our course assignment environment where preferences are more heterogeneous.

Second, courses need a minimum number of attendees. Minimum quotas are also an issue for college admission (Biró 2008), for the assignment of medical residents (Kamada and Kojima 2015), or in applications of the US Military Academy (Sönmez and Switzer 2013). Often, minimum quotas are also implemented as a means to guarantee diversification of quality or maintaining racial and ethnical balance (Abdulkadiroğlu and Sönmez 2003, Ehlers et al. 2014). Unfortunately, considering minimum quotas leads to a lot of additional complexity as we will show. We propose a course assignment mechanism based on TTC that is substantially fairer than TTC, remains strategyproof and efficient, and allows for minimum quotas.

1.2. TTC and Extensions

TTC was introduced in Shapley and Scarf (1974b) in order to find a competitive solution for a stylized housing market. In the Shapley-Scarf housing model, each person is endowed with a house. A round of TTC proceeds as follows. Each person points at her favorite remaining house, and each house points to its owner. There must exist at least one cycle. For each cycle, assign the person to the house she is pointing to and then remove the agent and house. The mechanism proceeds by repeating this process until no people or houses remain. The course assignment (and school choice) problem differs from the house assignment problem in two important ways: courses may be assigned to more than one student, and no student "owns" a course. Nonetheless, Abdulkadiroğlu and Sönmez (2003) adapt TTC to this environment in a natural way. Each student points to her favorite course and each course points to the student with highest priority at that course.

In the original formulation of TTC, a house points at its owner in order to preserve individual rationality. With course assignment, there is no individual rationality as there is no ownership. When there are no minimum capacities, a natural analog to individual rationality is respecting top priorities: if an object a has capacity for q agents, then each of the q highest ranked students at a either are assigned to a or a school they prefer to a. We say these students are guaranteed a. It is not necessary to point to the highest ranked student at a in order to preserve respecting top priorities. We can point to any of the guaranteed students. Note that if a student is guaranteed a and she ranks a first, then assigning her to a cannot result in justified envy. It is only when a student does not rank a first (and is part of a cycle) that it is possible for her assignment to cause justified envy. In this situation, the student's priority at a is irrelevant except to the extent that it guarantees her assignment at a. What is relevant is her priority at schools other than a. A designer cannot know which object an agent will point to (and it will not be strategyproof if it uses a student's submitted rankings). However, since the course priorities are known to the mechanism designer, a better approach is to point to the student that is most likely to have high priority at the courses she points to. We combine this with the "clinching" procedure introduced by Morrill (2013) to define an algorithm called Prioritized Clinch and Trade.

Our major technical challenge is determining which students are guaranteed a course. This is trivial when there is no minimum quota. With no minimum quota, an agent is guaranteed a course c with a capacity for q students if an only if she has one of the q highest priorities at c. It is far less clear when each course must be assigned a minimum number of students. In order to incorporate minimum quotas, we draw on a recent contribution by Fragiadakis et al. (2012) and Fragiadakis et al. (2016). The authors proposed extensions of the DA and TTC algorithms for school choice by dividing the number of available seats into two classes; regular seats which equal the minimum quotas and must be filled, and extended seats which equal the difference between the maximum and minimum quota of a school. The authors proposed, among others, the Extended-Seat DA and Extended-Seat TTC algorithms. While the former is group strategyproof and fair, the latter is strongly group strategyproof and Pareto efficient.

1.3. Contribution and Outline

In order to increase fairness of TTC, we propose three extensions for TTC as outlined in Figure 1: (1) clinching of guaranteed seats by students; (2) a prioritized pointing rule; and, (3) widening the range of guarantees in order to leverage the effects of the former two extension. The wider the range of guaranteed seats can be set, the bigger are potential benefits of clinching and prioritized pointing. Our proposed assignment mechanism is therefore called **E**xtended **S**eat **P**rioritized **C**linch and **T**rade with a widened **r**ange of guarantees (hereafter **RESPCT**).

We compare the performance of our algorithm to traditional TTC in the large-scale course assignment application at TUM. In all, we utilized ten different assignments ranging from 27 to 733 different students. Minimum quotas were originally not considered in these assignments (indeed, part of the motivation for our work was to determine how best to incorporate them), so we compared instances of the algorithm for a variety of minimum quota scenarios. In every scenario we considered, our algorithm was substantially

Figure 1 The three integral parts of our preferred assignment mechanisms, RESPCT

fairer than Extended-Seat TTC, resulting in a reduction of instances of justified envy by a factor of three or more. We also find substantial improvement along the number of students that envy other students.

In summary, our assignment is efficient and has far fewer students with justified envy compared to Extended-Seat TTC, and when a student is assigned unfairly, far fewer students have justified envy of her. Compared to Extended-Seat DA roughly 10 percent of the students could be Pareto-improved with RESPCT. Strategyproofness, efficiency and high levels of fairness convinced the Informatics department at TUM, i.e. the Dean of studies and the student union, to change the assignment procedure from DA to RESPCT.

The paper is structured as follows. In Section 2, we discuss work related to our problem. Afterwards, we define the course assignment model in Section 3. Since even without minimum quotas, TTC can lead to unfair assignments, we discuss ways to make the mechanism fairer in Section 4. Afterwards, we introduce our mechanisms for minimum quotas in Section 5 and provide a computational study in Section 6. Finally, we summarize our results and draw conclusions in Section 7.

2. Related Literature

TTC was originally introduced in Shapley and Scarf (1974b) while Roth (1982b) demonstrated that TTC is strategyproof. The first characterization of TTC for the housing model was provided by Ma (1994) who demonstrates that TTC is the unique strategyproof, Pareto efficient, and individually rational mechanism in the Shapley-Scarf housing market. However, the main applications of TTC are not in the housing market (where there is ownership) but instead in object allocation where the objects have priorities. Pápai (2000) is the first to modify TTC for object allocation as part of a broad class of allocation mechanisms called hierarchical exchange rules. This class was later generalized by Pycia and Ünver (2017) who introduce a generalization of hierarchical exchange rules called trading cycles. Much of the recent attention to TTC is due to Abdulkadiroğlu and Sönmez (2003). This pioneering paper demonstrated the applicability of

TTC to object allocation. A number of recent papers that have provided characterizations of TTC for this environment are Abdulkadiroğlu et al. (2017), Morrill (2013), Morrill (2015a), and Dur (2013).

There have been two other papers that have modified TTC with the intention of making it fairer. Most similar to our paper is Hakimov and Kesten (2018) which introduces the algorithm Equitable Top Trading Cycles (ETTC) which addresses the same question that we do; is it possible to make TTC fairer? Hakimov and Kesten (2018) compares the performance of their algorithm to TTC both in the lab and in simulations, and in both environments, they find their algorithm is substantially fairer than TTC. ETTC does, however, not allow for minimum quotas. The other paper to modify TTC with the intention of making it fairer is Morrill (2015b). This paper introduces the idea of "clinching" which does not allow a student who has already guaranteed assignment at a school to trade for that school. Clinching is incorporated in our preferred assignment algorithm.

Another closely related paper to ours is Abdulkadiroğlu et al. (2017). As far as we know, their paper is the only other paper that considers the performance of TTC versus alternatives using real-world data. They consider the problem of assigning children to schools in New Orleans where TTC was used for one year's assignment. They compare the performance of TTC versus alternative implementations and find essentially a negligible impact on the fairness of the assignment. School assignment and course assignment are closely related problems, but there are also substantial differences. There is far more heterogeneity in preferences in course assignment than in the typical school choice problem.[1] Moreover, the "depth" of the assignment problem, in the sense of the ratio of students to objects, is different as far more students are assigned to a school than to a course.

More generally, our paper is similar in spirit to Erdil and Ergin (2008) and Abdulkadiroğlu et al. (2009). These papers demonstrate that a seemingly innocuous feature of the deferred acceptance algorithm, how ties in a school's priority ranking are broken, can have a significant impact on the efficiency of deferred acceptance.

3. Model

3.1. Notation and Basic Algorithms

As a typical example for a matching market, we consider a course allocation problem as a tuple $(S, C, p, q, \succ_S, \succ_C)$. Here, $S = \{s_1, s_2, ..., s_n\}$ is the set of students, while $C = \{c_1, c_2, ..., c_m\}$ is the set of courses. The vector

[1] For example, one of the authors is part of the team that designed and administers the magnet school assignment for Wake County Public Schools, the 16^{th} largest school district in the United States. For this assignment there are no neighborhood schools, and the typical student has identical priorities at every school. The only exception is if they have an older sibling already attending one of the schools.

$p = (p_1, p_2, ..., p_m)$ describes the courses' minimum quotas, while $q = (q_1, q_2, ..., q_m)$ is the vector of their maximum quotas (capacity). We assume that $p_c \le q_c$ for all c and $\sum_{c\in C} p_c \le n \le \sum_{c\in C} q_c$.

Each student $s \in S$ has a complete, irreflexive, and transitive preference relation $\succ_s$ over $C \cup \{\circ\}$ where $\circ$ represents a student being unassigned and $q_\circ = \infty$. Then, $a \succ_s b$ indicates that s strictly prefers course a to b. Correspondingly, each course $c \in C$ has a strict, complete, irreflexive and transitive preference relation $\succ_c$ over students. Let $\mathcal{P}^{|S|}$ be the set of all possible preference profiles for all students. A matching $\mu : C \cup S \mapsto 2^S \cup C$ is a function which satisfies the following consistency conditions: $\mu(s) \in C$ for all $s \in S$, $\mu(c) \subseteq S$ for all $c \in C$ and $\mu(s) = c$ if and only if $s \in \mu(c)$ for all $s \in S$, $c \in C$.

TTC was developed by Shapley and Scarf (1974a) as the unique solution which is at the core of the house allocation problem. The one-to-many generalization was provided by Abdulkadiroğlu and Sönmez (2003). The mechanism is represented by Algorithm 1.

Algorithm 1 Top Trading Cycle

Assign a capacity counter for each course which keeps track of how many seats are still available at the course. Initially set the counters equal to the maximum quotas of the courses.

Round $k \ge 1$. Each student points to her favorite course with remaining capacity. Each course points to the student who has the highest priority for the course. Since the number of students and courses are finite, there is at least one cycle. Moreover, each student and each course can be part of at most one cycle. Every student in a cycle is assigned a seat in a course she points to and is removed. The counter of each course in a cycle is reduced by one and if it reduces to zero, the course is removed.

Extended Seat Top Trading Cycle (ESTTC) is a mechanism proposed by Fragiadakis et al. (2012) to incorporate minimum quotas for matching markets. ESTTC works as follows: for the regular matching market $(S, C, p, q, \succ_S, \succ_C)$ described above define an extended market $(S, \tilde{C}, \tilde{q}, \tilde{\succ}_S, \tilde{\succ}_{\tilde{C}})$, where $\tilde{C}$ can be derived from C by dividing every course $c \in C$ into a standard course c with maximum quota $\tilde{q}_c = p_c$ and an extended course c^* with maximum quota $\tilde{p}_{c^*} = q_c - p_c$. In the extended market, there are no more minimum quotas. We use the same notation for both regular courses and standard courses in the extended market with a slight abuse of notation. $\tilde{\succ}_S$ is obtained by inserting every extended course c^* in every student's preference profile directly behind the corresponding standard course. Thus, if $\succ_s = c_i \succ_s c_j \succ_s ...$ for some $s \in S$, $\tilde{\succ}_s = c_i \tilde{\succ}_s c_i^* \tilde{\succ}_s c_j \tilde{\succ}_s c_j^* ...$ in the extended market. In the remainder of this work we will only consider these extended courses (and their pointing rules within the mechanism) when the corresponding standard course has no more capacity and is hence removed from the market. Let $\epsilon = n - \sum_{c\in C} p_c$ be the number of students which can at maximum be assigned to extend courses without making a feasible matching impossible. In addition to the individual preferences of courses, there also exists an additional global preference list $\succ_{ML}$ (master

---

Algorithm 2 Extended Seat Top Trading Cycle

Set the counter of available seats for all (standard and extended) courses to their respective maximum quotas.

Round $k \geq 1$. Each student points to her favorite course with remaining capacity, each standard course points to her favorite student and each extended course points to the student with the highest priority according to $\succ_{ML}$. There must exist at least one cycle. Every student in a cycle is assigned a seat in a course she points to and is removed. The counter of each course in a cycle is reduced by one and if it reduces to zero, the course is also removed. If the number of students assigned to extended courses reaches ϵ after a round, remove all extended courses.

---

list), which Fragiadakis et al. (2012) intuitively describe as a type of tiebreaker when two students cannot both be assigned their preferred courses. ESTTC is then defined as in Algorithm 2.

Denote $\tilde{\mu}$ the matching in the extended market. Then, the matching μ in the regular market is defined as follows: $s \in \mu(s)$ if and only if $s \in \tilde{\mu}(c)$ or $s \in \tilde{\mu}(c^*)$. Since all extended courses point to the same student, at most one student can be assigned an extended course in every round. Therefore, ESTTC always results in a matching which adheres to the minimum quotas. However, this strict policy of endowing all seats to a single student is one of the major shortcomings of ESTTC and can be relaxed in order to make the mechanism fairer as we describe in Section 4.

Example 1. Let $s = \{s_1, ..., s_6\}$ and $C = \{c_1, c_2, c_3\}$. The preferences, priorities and quotas are provided in Table 1. The rounds of the mechanism are depicted in Figure 3.1. To keep the example simple, the number of students matches the sum of maximum quotas and therefore the number of students assigned to extended schools can never exceed $\epsilon = n - \sum_{c \in C} p_c = 6 - 3 = 3$.

Round 1: Standard courses c_1 and c_2 point to their favorite student while extended course c_3^* points at s_1 which is highest in the ML-list. There exists only one cycle: $c_1, s_3, c_3^*, s_1, c_1$. We resolve the cycle by setting $\mu(c_1) = s_3$ and $\mu(c_3^*) = s_3$ and reducing the capacities of c_1 and c_3^* by one. We remove c_3^* as its remaining capacity reaches 0.

Round 2: We obtain c_1, s_5, c_2, s_2, c_1 as only existing cycle in round 2 and resolve that cycle. After resolving the cycle, we remove c_1 and c_2.

Round 3: We resolve the cycle c_1^*, s_4, c_1^*. After that, we remove c_1^*.

Round 4: We resolve the cycle between the only remaining course c_2^* and the only remaining student s_6.

We obtain the matching $\mu' = \{(c_1, \{s_1, s_2\}), (c_1^*, \{s_4\}), (c_2, \{s_5\}), (c_2^*, \{s_6\})\}, (c_3^*, \{s_3\})\}$ in the extended market which results in the matching $\mu = \{(c_1, \{s_1, s_2, s_4\}), (c_2, \{s_5, s_6\}), (c_3, \{s_3\})\}$ in the regular market.

s_1	s_2	s_3	s_4	s_5	s_6
c_1	c_1	c_3	c_3	c_2	c_3
c_3	c_2	c_2	c_1	c_1	c_1
c_2	c_3	c_1	c_2	c_3	c_2

	c_1	c_2	c_3	ML
	s_3	s_2	s_1	s_1
	s_5	s_1	s_6	s_2
	s_1	s_3	s_4	s_3
	s_6	s_6	s_5	s_4
	s_2	s_4	s_2	s_5
	s_4	s_5	s_3	s_6
p	2	1	0	
q	3	2	1	

Table 1 Preferences and priorities for student-to-course matching with minimum quotas used for the examples in this paper

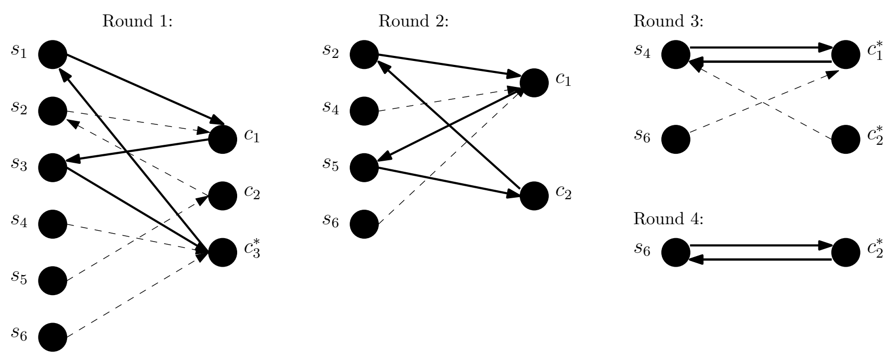


Figure 2 Run of ESTTC for Example 1

3.2. Design Desiderata

A matching is called feasible if it satisfies the following property: $p_c \leq |\mu(c)| \leq q_c$ for all $c \in C$. Let $\mathcal{M}$ denote the set of feasible matchings. A mechanism $\chi : \mathcal{P}^{|S|} \mapsto \mathcal{M}$ is a function which takes a set of preference profiles as input and returns a feasible matching between students and courses. We assume the courses' priorities to be fixed and known to all students. Ideally, we want students to have incentives for reporting their true preferences to the mechanism.

Definition 1 (Strategyproofness). A mechanism χ is strategyproof if and only if for any $\succ_S \in \mathcal{P}^{|S|}$, $s \in S$ and $\succ'_s \in \mathcal{P}$, $\chi_s(\succ_S) \succ_s \chi_s(\succ_{S\setminus\{s\}}, \succ'_s)$, where $\chi_s(\succ_S)$ is the matching that s receives under $\succ_S$.

Two main design desiderata for matching with preferences are Pareto efficiency and fairness.

Definition 2 (Pareto Efficiency). A matching μ is Pareto efficient if and only if no different feasible assignment μ' exists such that $\mu'(s) \succ_s \mu(s)$ or $\mu'(s) = \mu(s)$ for all $s \in S$ and $\mu'(s) \succ_S \mu(s)$ for some $s \in S$. If μ' exists, we say that μ' *Pareto dominates* μ.

Definition 3 (Fairness). A matching μ is fair if and only if $\mu(s') \succ_s \mu(s)$ implies $s' \succ_{\mu(s')} s$ for all $s, s' \in S$. If $\mu(s') \succ_s \mu(s)$ and $s \succ_{\mu(s')} s'$ for some s', we say that s and c form a blocking pair or s has *justified envy* towards s'.

Example 1 contains 4 instances of justified envy: s_6 has justified envy towards s_2, s_3 and s_4 and s_4 has

s_1	s_2
c_1	c_1
c_2	c_2

	c_1	c_2
	s_1	s_2
	s_2	s_1
p	0	1
q	2	2

Table 2 Example of a mutual best violation

justified envy towards s_3.

It is well known that for two-sided matchings, these two desiderata are incompatible and one has to choose between either Pareto efficiency or fairness. The DA mechanism by Gale and Shapley (1962) is strategyproof and fair while TTC is strategyproof and Pareto efficient (Shapley and Scarf 1974b). We aim for a mechanism that is strategyproof, efficient, and as fair as possible. We interpret mechanism A to be fairer than mechanism B for a given assignment problem if on average there are fewer instances of justified envy under A than under B.

A far weaker fairness condition is *mutual best*:

Definition 4 (Mutual Best). A matching μ satisfies mutual best if and only if the following condition is satisfied: c is the favorite course of s and s is among the q_c highest priorities of c implies that $\mu(c) = s$. If s is among the q_c highest priorities of c, we say that *s is guaranteed a seat at c*.

TTC satisfies mutual best; if s is guaranteed a spot at her most preferred course c, then s is always assigned to c. Note that not all mechanisms used in practice satisfy mutual best. For example, a serial dictatorship and a linear programming procedure do not satisfy mutual best. See Morrill (2013) for discussions on how mutual best relates to TTC. Unfortunately, in general, mutual best is incompatible with minimum quotas.

Theorem 1. *A matching which satisfies mutual best does not always exist in the presence of minimum quotas.*

Proof: Consider an example with $S = \{s_1, s_2\}$ and $C = \{c_1, c_2\}$ and preferences as in Table 2:

Either s_1 or s_2 is not assigned c_1, which is a contradiction to mutual best. Q.E.D.

In the following, we define a relaxation of mutual best.

Definition 5. A matching μ satisfies σ-mutual best if and only if the following condition is satisfied: (c is the favorite course of s and s is among the σ_c highest priorities of c) implies that $\mu(c) = s$.

Given this definition, mutual best is equivalent to σ-mutual best with $\sigma = q$. As shown above, in the presence of minimum quotas, this is not always achievable. The relaxation of mutual-best in the form of σ-mutual best is similar to the technique introduced in Fragiadakis et al. (2012) where the authors introduce σ-fairness as a relaxation of fairness.[2]

[2] In σ-fairness, as introduced by Fragiadakis et al. (2012), under certain conditions blocking pairs of s and c can be ignored if $|\mu(c)| > \sigma_c$

In this paper, we will discuss how to guarantee as many seats as possible, i.e. maximizing σ. Afterwards, we will introduce a mechanism, RESPCT, that satisfies σ-mutual best for this maximal vector of guaranteed seats and show empirically that this mechanism is substantially fairer than ESTTC.

4. Making Top Trading Cycles Fairer

TTC is both strategyproof and Pareto efficient (Roth 1982b). To see why TTC is efficient, note that each student in a cycle receives her most preferred course among those that have not already been assigned. Therefore, we cannot improve a student's assignment without harming a student that has been previously assigned. TTC is strategyproof for two reasons. First, a student's report does not effect which student a course points to. Second, a student has no incentive to create a cycle early. The pointing in TTC induces a directed graph in a natural way.[3] If, in any round of TTC, there is a path from a course c to a student s, then s has the ability to be assigned to c by pointing at c. However, it is never in her benefit to point to c until it is her favorite course because once there exists a path from c to s, that path remains until s is assigned. When s is assigned, she receives her most preferred course among those with available capacity, so by revealed preference, she must weakly prefer her assignment to c.

In the one-to-many generalization of TTC by Abdulkadiroğlu and Sönmez (2003), in each round, each student points to her most preferred remaining course, and each course with available capacity points to the remaining student with highest priority. Our approach to making TTC fairer leverages two observations regarding the one-to-many variant of the assignment problem.

First, the pointing rule of TTC is myopic. TTC always points at the highest ranked student, but we can point to any of the guaranteed students at a course and preserve the properties of efficiency, mutual best, and strategyproofness. Consider a course c. The question is which of the guaranteed students should c point to in order to minimize the chance of generating justified envy. The important point is that a student's priority at c is irrelevant to the probability she will cause justified envy. A cycle in TTC is either trivial, where a student and course point at each other, or nontrivial. A trivial cycle cannot cause justified envy so long as the student was guaranteed that course. In a nontrivial cycle, a student is not assigned to the course pointing at her, so her priority at that school is irrelevant.

Consequently, a better approach is to estimate which student is least likely to point at a course where she has low priority. Note that we cannot use a student's submitted preferences to determine who she will point to without violating strategyproofness. However, the designer does know a student's priority rank at all of the other courses.

In theory, one could estimate the likelihood that a course pointing to a student will cause justified envy. However, this is not computationally tractable. We would need an estimate of the students preferences (who is she likely to point to); an estimate of the probability that a student pointing to a course will form a cycle;

[3] Formally, the vertices of the graph consist of students and courses. There is a directed edge from a student (course) to a course (student) if the student (course) points to the course (student).

and an estimate of whether the demand for a course will exceed supply (i.e. will other students envy her assignment to the course). Rather than try to directly estimate this, we use a proxy which incorporates these factors. First, we restrict a course c with capacity q to only point to one of the q highest ranked students. This is to ensure that trivial cycles cannot cause justified envy. Second, among this set of students, we do not use a student's rank at c as this information is irrelevant. Instead, we calculate each students average rank at the other courses other than c. In this context, we use the term *highest average rank* for notational convenience but in fact we use a more sophisticated metric. We weight ranks at courses such that courses with a low remaining capacity are given high precedence while courses with a large remaining capacity are given low precedence. The relative capacity of the remaining courses is a proxy for the likelihood that a course will overdemanded. As processing a cycle changes the relative capacities of the remaining courses, these weights are dynamically adjusted.

Our second observation is that there are unnecessary trades in TTC which cause fairness distortions without contributing to TTCs properties Pareto efficiency, strategyproofness and mutual best. Consider the following example:

Example 2. Let $S = \{s_1, s_2, s_3\}$ and $C = \{c_1, c_2\}$. $q_{c_1} = 2$ and $q_{c_2} = 1$. Define preferences and priorities as follows:

s_1	s_2	s_3
c_2	c_1	c_2
c_1	c_2	c_1

c_1	c_2
s_1	s_2
s_2	s_3
s_3	s_1

In round one, we resolve the cycle c_2, s_2, c_1, s_1, c_2. In round two, we assign the only remaining student and course c_1, s_3. The final matching is $\mu = \{(c_1, \{s_2, s_3\}), (c_2, \{s_1\})\}$.

A matching which is both fair and Pareto efficient exists in the market: $\{(c_1, \{s_1, s_2\}), (c_2, \{s_3\})\}$. However, this matching is not chosen by TTC. Resulting from mutual best, s_2 always has to receive c_1 since c_1 is her favorite course and she has one of the q_{c_2} highest priorities at c_2. Under TTC, s_2 has to trade endowments with s_1 to receive a seat at c_2, thereby allowing s_1 to obtain a seat at c_2 where she has a low priority. This observation was first made in Morrill (2013) and the following modification is the basis for his variant of TTC called Clinch and Trade. For all students $s \in S$ and $c \in C$, whenever the priority of s at c is within the remaining capacity of c and c is the favorite course of s, *clinch* s and c, i.e. match s with c, take s off the market and reduce the remaining capacity of c by one.

In the following we describe the Prioritized Clinch & Trade (PCT) mechanism, which corporates both ideas, a more sophisticated pointing of the courses as well as a clinching procedure:

Algorithm 3 Prioritized Clinch and Trade

Round $k \geq 1$ - Clinching: Whenever a student s who is not pointing to a course at the end of round $k-1$ (i.e., the course s is pointing to in round $k-1$ is removed in round $k-1$) has a priority which is within the remaining capacity of her favorite course c, assign s to c. Iteratively clinch until no student-course pair can be clinched.

Round $k \geq 1$ - Trading: Let each student s point to her favorite course with remaining capacity and let each course c who was not pointing by the end of round $k-1$ point to the student who is guaranteed a seat at c with the highest average rank among courses other than c. There must exist at least one cycle. Resolve the cycle by assigning each student the course she is pointing to. After that, remove every assigned student, reduce the capacity of each course in the cycle by one, and remove courses with capacity of zero.

THEOREM 2. *PCT is Pareto efficient, strategyproof and satisfies mutual best.*

We will provide an example of the clinch and trade mechanism in the next section, when discussing the extension of PCT for course assignments with minimum quotas.

It is well known that it is impossible for a mechanism to be strategyproof, fair, and always select an efficient assignment when a fair and efficient assignment exists. However, there does exist a strategyproof, fair, and efficient mechanism when there are only two schools and their is sufficient capacity for all students: PCT. A fair and efficient assignment must correspond to the student-optimal stable assignment, so this also implies that when there are two schools and sufficient capacity, that PCT corresponds exactly to the student-proposing Deferred Acceptance algorithm. Rather interestingly, when there are N students, N schools, and each school has a capacity of one, then PCT is equivalent to TTC. However, when there are N students, two schools, and the total capacity of both schools is at least N, then PCT is equivalent to the Deferred Acceptance algorithm.

THEOREM 3. *If there are two schools and the total capacity of the two schools is greater than or equal to the number of students, then PCT is fair.*

Note that TTC is not fair when there are two schools and the schools have sufficient capacity. When there are more than two schools, PCT may have justified envy. However, the intuition from Theorem 3 continues to hold more generally. When there are fewer schools, average priority becomes a better predictor of a student's priority at her favorite school. While PCT consistently outperforms TTC, the difference is most dramatic when their are relatively few schools with relatively large capacities.

COROLLARY 1. *If there are two schools and the total capacity of the two schools is greater than or equal to the number of students, then PCT corresponds exactly to the student-proposing Deferred Acceptance algorithm.*

5. Considering Minimum Quotas

The main design challenge we faced in creating the course assignment mechanism at TUM was how to incorporate minimum quotas. The two innovations we have introduced, clinching and non-myopic pointing, are based on identifying which students are guaranteed a school. This is trivial when there are no minimum quotas; a student is guaranteed a school with capacity q if and only if she has one of the q highest priorities. This is far more challenging when there are minimum quotas.

The ESTTC algorithm, introduced by Fragiadakis et al. (2012) and defined on page 8, is a hierarchical exchange rule as introduced in Pápai (2000). The inheritance rule is a hybrid between the standard TTC inheritance rule and that of a serial dictatorship. For a school c with minimum quota p_c and maximum quota q_c, the first p_c seats are inherited via TTC and the remaining seats are inherited as in a serial dictatorship using the ordering induced by the master list. When viewed this way, it is clear that ESTTC guarantees the assignment of p_c students at school c. Using this observation, ESTTC, can be adapted to PCT in a natural way. We refer to the resulting mechanism as Extended Seat Prioritized Clinch & Trade (ESPCT):

Algorithm 4 Extended Seat Prioritized Clinch and Trade (ESPCT)

Set $\epsilon = n - \sum_{c \in C} p_c$. Denote G_c the set of students who have one of the p_c highest priorities at course c. Instead of using an arbitrary master list, we define $\succ_{ML}$ using a student's average priority at all courses. Specifically, $i \succ_{ML} j$ if i has a higher average priority at all courses than j.

Round $k \geq 1$ - Clinching: Whenever a student s who is not pointing to a course at the end of round $k-1$ (i.e., the course s is pointing to in round $k-1$ is removed in round $k-1$) has a priority which is within the remaining capacity of her favorite standard course c, assign s to c. Iteratively clinch until no student-course pair can be clinched.

Round $k \geq 1$ - Trading: Let each student s point to her favorite course with remaining capacity. Let all standard courses which were not pointing to a student by the end of round $k-1$ point to the student in G_c with the highest average priority at course other than c. Let all extended courses point to the student ranked highest according to $\succ_{ML}$ (i.e., the student with the highest average priority at all courses). Execute one round of TTC. If the number of assigned extended seats reaches ϵ, remove all extended courses. Update G_c according the remaining students.

Example 3. We use the same instance as Example 1 with the corresponding preferences and priorities shown in Table 1 to allow for a comparison between ESTTC and ESPCT. We compute $\epsilon = 6 - 3 = 3$.
Round 1: No student clinches at the beginning of the round. Standard course c_1 points at s_3 due to her average rank of 4.5 among other courses (as compared to 5 of s_5) and standard course c_2 points at s_2. Extended course c_3^* points at s_1 due to her average rank of 2 across all courses. This leads to a cycle

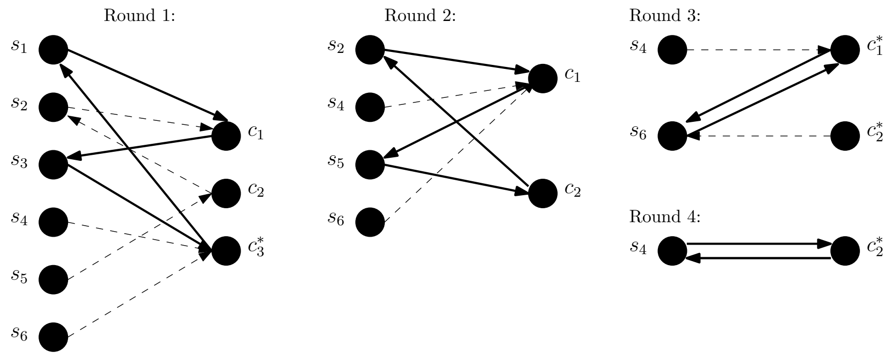


Figure 3 Run of ESPCT for Example 3

$c_1, s_3, c_3^*, s_1, c_1$ and c_3^* is taken off the market.

Round 2: Again, no student clinches. We resolve the cycle c_1, s_5, c_2, s_2, c_1. After that, we take c_1 and c_2 off the market.

Round 3: All extended courses point to s_6. We resolve c_1^*, s_6, c_1^*.

Round 4: We resolve the cycle between the only remaining student s_4 and the only remaining course c_2^*.

The trading rounds are depicted in Figure 5. We receive the matching $\tilde{\mu} = \{(c_1, \{s_1, s_2\}), (c_1^*, \{s_6\}), (c_2, \{s_5\}), (c_2^*, \{s_4\}), (c_3, \{s_3\})\}$ in the extended market. The resulting matching in the regular market is $\mu = \{(c_1, \{s_1, s_2, s_6\}), (c_2, \{s_4, s_5\}), (c_3, \{s_3\})\}$. With s_6 and s_4 having justified envy towards s_3, μ contains 2 instances of justified envy.

First, note that the original version of ESTTC already satisfies p-mutual best (i.e. σ-mutual best with $\sigma = p$). Using this result, we can prove the same for ESPCT.

THEOREM 4. *ESTTC is strongly group-strategyproof and Pareto efficient and satisfies p-mutual best.*

COROLLARY 2. *ESPCT is strategyproof and Pareto efficient and satisfies p-mutual best.*

Proof: The proof is a simple consequence from Theorems 2 and 4. Q.E.D.

ESTTC (and ESPCT) guarantees the priority of only p_c students at course c (where p_c is c's minimum quota). But in general, this is not the maximum number of students it could guarantee. For example, consider a feasible minimum priority vector p and an alternative minimum priority vector p' where for every school c $p'_c \le p_c$ and for some school $p'_c < p_c$. It is clearly feasible to satisfy the minimum quotas p' (since it was possible to satisfy the quotas p). But note that if we run ESTTC using p', then fewer students are guaranteed any individual course than if we run ESTTC using p. For this reason, ESTTC will perform better (with respect to justified envy) under p than under p'. This is paradoxical as the minimum number of students required is a design *constraint* and is chosen to be as small as possible. This occurs because the algorithm transitions earlier from pointing to students based on priorities to pointing to students based on the master list.

We correct this by first determining a maximal number of seats that can be guaranteed at a school. This can be applied to any of the extended seat algorithms, but it is the most beneficial when we also us this information for clinching and prioritized pointing. We refer to this mechanism as **r**ange-widening ESPCT (RESPCT).

5.1. Maximizing the number of guaranteed seats

In order to allow courses to guarantee seats in excess of their minimum quotas, we need to make sure that these guarantees do not lead to a shortage of students for other courses in the case that all guaranteed seats are clinched. For each course c, let σ_c be the number of students that are guaranteed a seat at c, and G_c with $|G_c| = \sigma_c$ be the set of these students. Obviously, $\sigma_c \le q_c$ has to hold for all courses. Further, in order to be compatible with the minimum quota vector p, for each subset of courses C_0, the number of students that are not guaranteed seats in courses in C_0 must exceed the sum of minimum quota of other courses, i.e.

$$n - |\bigcup_{c\in C_0} G_c| \ge \sum_{c\in C\setminus C_0} p_c \text{ for all } C_0 \subset C \tag{FC}$$

has to hold. By satisfying this condition, we ensure that even if all seats guaranteed by courses in C_0 are clinched, there are still enough students left to satisfy the minimum quotas of all other courses.

In the following, we discuss how to maximize the number of guaranteed seats while retaining the feasibility condition (FC). Note first that we can always guarantee at least n seats.

Theorem 5. *Let σ be a vector of guaranteed seats with $\sum_{c\in C} \sigma_c = n$ and $p_c \le \sigma_c \le q_c$. Then,* (FC) *is satisfied.*

Under certain conditions, we can even guarantee $\sigma = q$:

Theorem 6. *If $n \ge \sum_{c'\in C\setminus\{c\}} q_{c'} + p_c$ for all $c \in C$, then* (FC) *is satisfied for $\sigma = q$.*

In general, checking whether a vector of guarantees σ is compatible with the minimum quota vector p, requires validating (FC) for an exponential number of subsets. The following result implies that it is not possible in general to efficiently show compatibility.

Theorem 7. *Deciding whether a vector σ is incompatible with a minimum quota vector p is a strongly NP-complete problem.*

The compatibility of a fixed vector σ can be verified via a mixed integer linear program. In this case, the parameter σ_{cs} denotes if s is guaranteed a seat at c in σ (i.e. $\sigma_{cs} = 1$ if and only if s is among the top σ_c students in c's preference list). Define variables $x_{sc} \in \{0, 1\}$ which states whether student $s \in S$ clinches her seat at $c \in C$. Further, introduce an auxiliary variable $\lambda_c \in \mathbb{R}_{\ge 0}$ for each course $c \in C$ to denote the shortage of students of c, i.e. the number of students that are necessary in excess to the students clinching a seat at c in order to satisfy the minimum quota p_c. Then, the objective of the following MIP is to find a clinching of seats by the students such that (FC) is violated. Thus, if the MIP does not allow for a feasible solution, (FC) is satisfied for the vector σ.

$$\sum_{c\in C} x_{sc} \leq 1 \qquad s \in S \tag{1}$$

$$\sum_{c\in C} x_{sc} \leq \sigma_{cs} \qquad s \in S, c \in C \tag{2}$$

$$\lambda_c \leq q_c(1 - z_c) \qquad c \in C \tag{3}$$

$$\lambda_c \leq p_c - \sum_{s\in C} x_{sc} + q_c z_c \qquad c \in C \tag{4}$$

$$\sum_{c\in C} \lambda_c \geq \sum_{s\in S}\left(1 - \sum_{c\in C} x_{sc}\right) + 1 \tag{5}$$

$$x_{cs} \in \{0, 1\} \qquad c \in C, s \in S \tag{6}$$

$$\lambda_c \geq 0 \qquad c \in C \tag{7}$$

$$z_c \in \{0, 1\} \qquad c \in C \tag{8}$$

Here, constraints (1) and (2) restrict each student to clinch at most one of her guaranteed seats. All students, which either do not clinch a guaranteed seat (or are not offered one in the first place by any course) are summed up in constraint (5) in which we enforce a solution such that the shortage of students exceeds the sum of non-clinching students. The shortage λ_c for each course c is defined in constraints (3) and (4), setting it to to $\max\{0, p_c - \sum_{s\in C} x_{sc}\}$.

Unfortunately, it is not easy to make use of the MIP in order to maximize the number of guaranteed seats. One could think of making the guarantees σ_{cs} variable and minimizing the sum $N = \sum_{c\in C}\sum_{s\in S} \sigma_{cs}$ of guarantees, i.e. finding the minimal number of guarantees such that the vector σ is incompatible with the minimum quotas. While finding such an N ensures that all vectors σ' with $\sum_c \sigma'_c < N$ respect (FC), this does not necessarily yield a maximal number of guaranteed seats. One approach is to embed the MIP within a bilevel optimization problem, however these problems are very computationally demanding to solve.

Another possibility to find vectors σ with the help of the MIP described above is by employing Algorithm 5. Starting with $\sigma = p$, in each step of the algorithm one element of σ is increased and the feasibility of the new vector is validated with the MIP. The algorithm terminates when σ can no longer be increases. This way, after at most $\sum_{c\in C} q_c$ iterations, the algorithm returns a maximal vector σ, i.e. for none of the courses its number of guaranteed seats can be increased. Note, that the algorithm does not necessarily find a maximum vector, i.e. one where the sum of guaranteed seats is maximized.

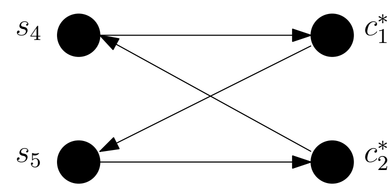


Figure 4 **Run of RESPCT for Example 4**

Algorithm 5 Greedy algorithm to determine a maximal σ vector

Set $\sigma = p$. Further, define a set Γ of courses c, for which σ_c cannot be increased. Initially, Γ includes only courses c with $\sigma_c = q_c = p_c$.

While $\Gamma \neq C$: Select a course $c \in C \setminus \Gamma$ and define σ' with $\sigma'_c = \sigma_c + 1$ and $\sigma'_{c'} = \sigma_{c'}$ for all $c' \neq c$. Solve the MIP for σ'. If the MIP returns a feasible solution (i.e the vector σ' violates (FC)), add c to Γ, otherwise set $\sigma = \sigma'$ and continue.

Return σ.

5.2. Range-Widened Extended Seats Prioritized Clinch and Trade

As a result of these considerations, we define the range-widened Extended Seats Prioritized Clinch and Trade (RESPCT) mechanism by widening the clinching and prioritization rules to also apply to extended courses according to the σ determined above. Note that σ is not fixed, and instead, we recalculate σ at the end of each round. We allow students to clinch with extended courses c^* as long as they have one of the σ_c highest priorities at c. Then, in the cycle resolution phase, we allow extended courses to point up to ϵ distinct students. For this, we iteratively select the courses and determine which students they point to. First, we consider all extended courses c^* with $\sigma_c > p_c$ and allow them to point at the student which has a guaranteed seat at c with highest average priority at other courses (as in the prioritized pointing rule in PCT). If these courses point to fewer than ϵ distinct students, we continue with the remaining extended courses which are allowed to point to their highest priority student. As soon as the number of students which are pointed at by the extended courses reaches ϵ, all remaining extended courses point at the student among those ϵ with highest priority.

Example 4. Again, we consider the same market from Examples 1. Since every matching which satisfies the maximum quotas is feasible, we get $\sigma = q$. Again, $\epsilon = 3$.

Round 1: s_1 clinches c_1. After that, s_6 clinches c_3^*. s_3 clinches her favorite remaining school c_2. s_2 then clinches her favorite school c_1. With only two students and two courses remaining, we resolve cycle $c_1^*, s_5, c_2^*, s_4, c_1^*$ as depicted in Figure 4.

We obtain the matching $\tilde{\mu} = \{(c_1, \{s_1, s_2\}), \{(c_1^*, \{s_4\}), (c_2, \{s_3\}), (c_2^*, \{s_5\}), (c_{3*}, \{s_6\})\}$, resulting in $\mu = \{(c_1, \{s_1, s_2, s_4\}), (c_2, \{s_3, s_5\}), (c_3, \{s_6\})\}$ in the regular market. μ contains no instance of justified envy.

Theorem 8. *RESPCT satisfies strategyproofness, Pareto efficiency and σ-mutual best such that $\sum_{c \in C} \sigma_c \geq n$.*

Algorithm 6 Range-widened Extended Seats Prioritized Clinch and Trade (RESPCT)

Determine σ^1 according to one of the methods described in Section 5.1. Set $\epsilon = n - \sum_{c\in C} p_c$. As with ESPCT, we define $\succ_{ML}$ using a student's average priority at all courses. Specifically, $i \succ_{ML} j$ if i has a higher average priority at all courses than j.

Round $k \geq 1$ - *Clinching phase*: Whenever a student s who is not pointing to a course at the end of round $k-1$ (i.e., the course s is pointing to in round $k-1$ is removed in round $k-1$) has a priority which is within σ_c^k of her favorite course c, assign s to c. After each round of clinching, recalculate σ^k. Iteratively clinch until no student-course pair can be clinched.

Round $k \geq 1$ - *Cycle Resolution Phase*: All standard courses and extended courses which were pointing to a student by the end of round $k-1$ continue to point at that student. Each standard course c, which was not pointing to a student in round $k-1$, points to the student in G_c^k with the highest average priority at courses other than c. For each extended course c, with $\sigma_c^k > p_c^k$ and which was not pointing to a student by the end of round $k-1$, points to the student in G_c^k with the highest average priority at courses other than c. If all extended courses point to fewer than ϵ many students, iteratively select additional extended courses and have them point to their highest prioritized student.[4] If extended courses point to ϵ students already, let all other extended courses point to the student among those ϵ who has the highest priority according to $\succ_{ML}$ (i.e., the highest average priority at all courses). Execute one round of TTC. Recalculate σ, and for all c, set $\sigma_c^{k+1} := \sigma_c$. If the number of assigned extended seats reaches ϵ, remove all extended courses.

THEOREM 9. *If* $\sum_{c'\in C\setminus\{c\}} q_{c'} + p_c \geq n$ *for all c, RESPCT satisfies mutual best.*

Proof: This is a simple consequence of Theorems 6 and 8. Q.E.D.

6. Experimental Evaluation

In order to evaluate the mechanisms discussed in this paper, we conducted a computational study, using field data from the Informatics department at the Technical University of Munich (IN.TUM).

6.1. Data and Experimental Design

In the following, we briefly describe the field data from the course assignment application at IN.TUM. Students and lectures participate in this mechanism, students by stating their preferences over courses; lecturers by stating the capacity of their course and their preferences over students. The data sets describe preferences for the registration to seminars and practical courses for both bachelor and master students collected between June 2014 and March 2016. For our experiments, we only considered data sets where the number of students does not exceeds the total course capacity but does exceed the sums of minimum quotas we introduced. These sets range from 27 to 733 different students and 6 to 43 courses (see Table 3).

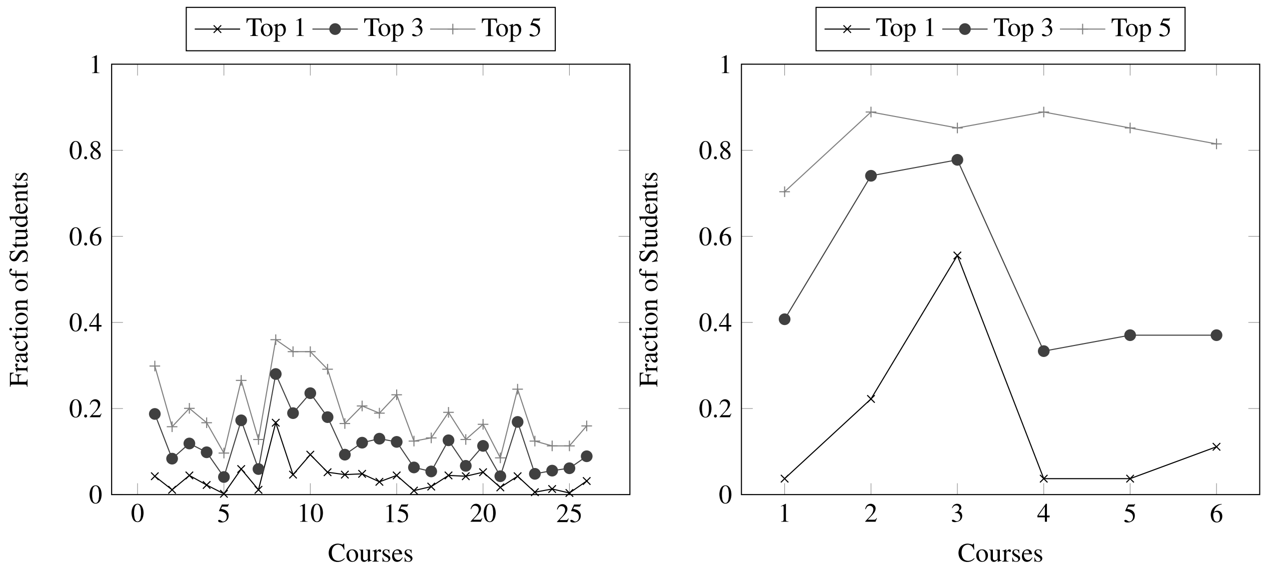
Table 3 Summary of Two-Sided data sets

Name	#Students	#Courses	Total capacity
TS1	539	26	575
TS2	27	6	59
TS3	88	12	116
TS4	689	36	726
TS5	57	9	86
TS6	636	40	758
TS7	105	11	110
TS8	78	16	253
TS9	731	40	775
TS10	733	43	753

Let us provide a few additional statistics on the distribution of preferences, highlighting their heterogeneity. For this, we illustrate the distribution of preferences in a larger data set, TS1, and a smaller data set, TS2, which are representative for other data sets as well. Figure 5 shows for each course in the two data sets the fraction of students that rank the corresponding course as their top choice, or among their top 3 (or 5) choice, respectively. The chart shows the heterogeneity of the students' preference lists except for a few popular courses, especially when considering the top 5 choices. Even in TS2 with only 6 courses, no course was ranked within top 5 by every student. The courses' priorities are equally diverse. In TS1, 15 students are ranked as top priority by at least one of the 26 courses, while 63 distinct students are ranked among the top 3 priorities and 106 students are ranked among the top 5, with a maximum number of $26 \cdot 3 = 78$ and $26 \cdot 5 = 130$ spots available. For TS2, 5 distinct students are ranked as top priority, 14 within top 3 and 18 within top 5, with a maximum of $6, 18$ and 30 spots available, respectively.

Course organizers submitted only maximum capacities since minimum quotas were not yet part of the course assignment mechanisms. Thus, in order to evaluate the mechanisms for each of the data sets we tested a range of different minimum quotas. We consider a range from 3 to 7 as reasonable minimum quotas for seminars and courses. Thus, for every $x \in [3, 7]$ we augmented each test set with minimum quotas $p_c = min(x, q_c - 1)$ for all courses c in the test set, resulting in a total of 50 test instances.

For all instances, we employed ESTTC, ESPCT, and RESPCT in order to compare their performance with respect to fairness. For this, we evaluated (1) the instances of justified envy, (2) the number of students with justified envy, and (3) the number of students envied. Moreover, in order to analyze the gain in efficiency when using a Pareto-optimal mechanism over a fair mechanism, we also implemented the extended-seats version of the deferred acceptance mechanism by Fragiadakis et al. (2016) to compare its efficiency to that of RSPECT, the fairest Pareto-optimal mechanism. Runtime is typically a criterion when evaluating algorithms. The longest runtime we observed was 139.03 seconds (2.32 minutes), while most algorithms took less than one minute, which is negligible in this domain.



(a) Data Set TS1 (26 courses)

(b) Data Set TS2 (6 courses)

Figure 5 **Fraction of students that have a course in the top 1,3 or 5 of their preference lists**

6.2. Fairness

In order to obtain the vector σ that is necessary for RESPCT, we first tested whether $\sigma = q$ was feasible. Only in 13 of the 50 test instances, $\sigma = q$ violated the feasibility condition (FC). For these, we employed Algorithm 5 to find a maximal vector of guarantees, $\sigma^{\max}$. In spite of having to solve several integer programs using Gurobi 7.0.2, finding $\sigma^{\max}$ only took a couple of seconds.

In the following, we discuss the degree of (un-)fairness of our mechanisms. For ESTTC, we used a randomized master list. In order to reduce the impact of chance we performed 10 runs of ESTTC for every minimum quota vector and test set with different master lists and computed the arithmetic mean of all metrics.

Figure 6 shows for each minimum quota vector the *instances of justified envy per student*, averaged over all ten instances. RESPCT performed best among all mechanisms, even with a lower number of guaranteed seats. It also becomes clear that even without guaranteeing more seats, the effects of clinching and trading in ESPCT led to much fairer assignments than the standard ESTTC.

There are many interesting effects that can be observed. First, as is expected, RESPCT leads to worse results as the minimum quotas increase. The minimum quotas are a constraint, and higher minimum quotas means that fewer students can be guaranteed a course. This reduces the impact of both clinching and prioritized pointing. Paradoxically, the performance of ESTCC and ESPCT actually improve as the minimum quotas increase. This was discussed earlier and is due to the fact that with higher minimum quotas, ESTTC and ESPCT allocate fewer extended seats. Since extended seats are assigned based on a the master list and not priorities at the course, having fewer extended seats results in less justified envy. Therefore, the performance difference between RESPCT and both ESTTC and ESPCT becomes smaller as the minimum quotas increase.

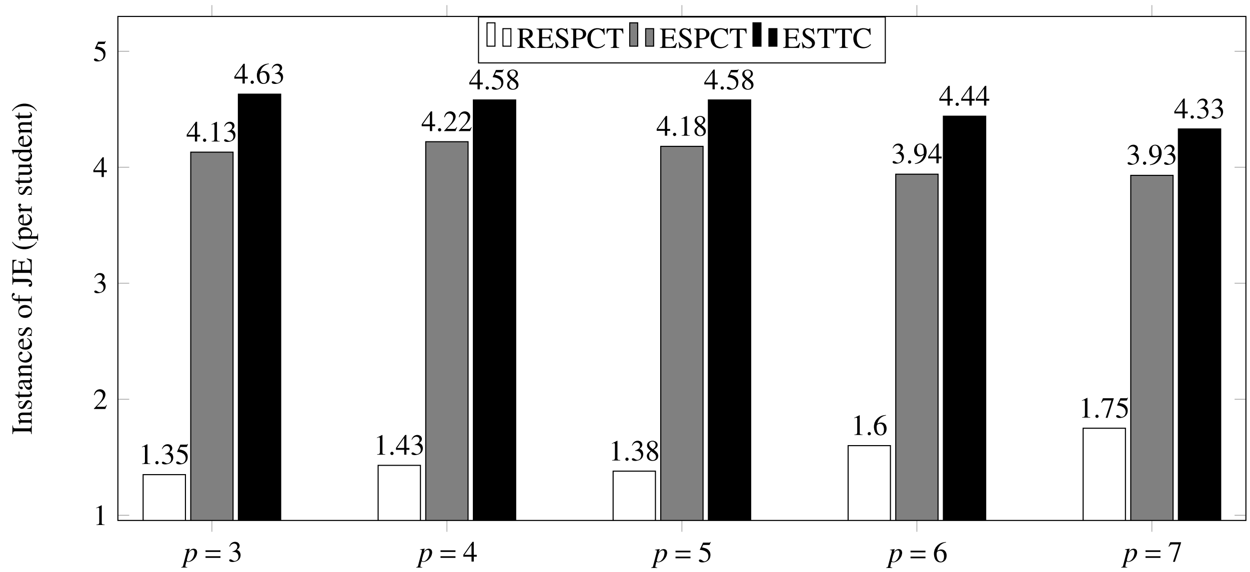


Figure 6 **Instances of justified envy (per student)**

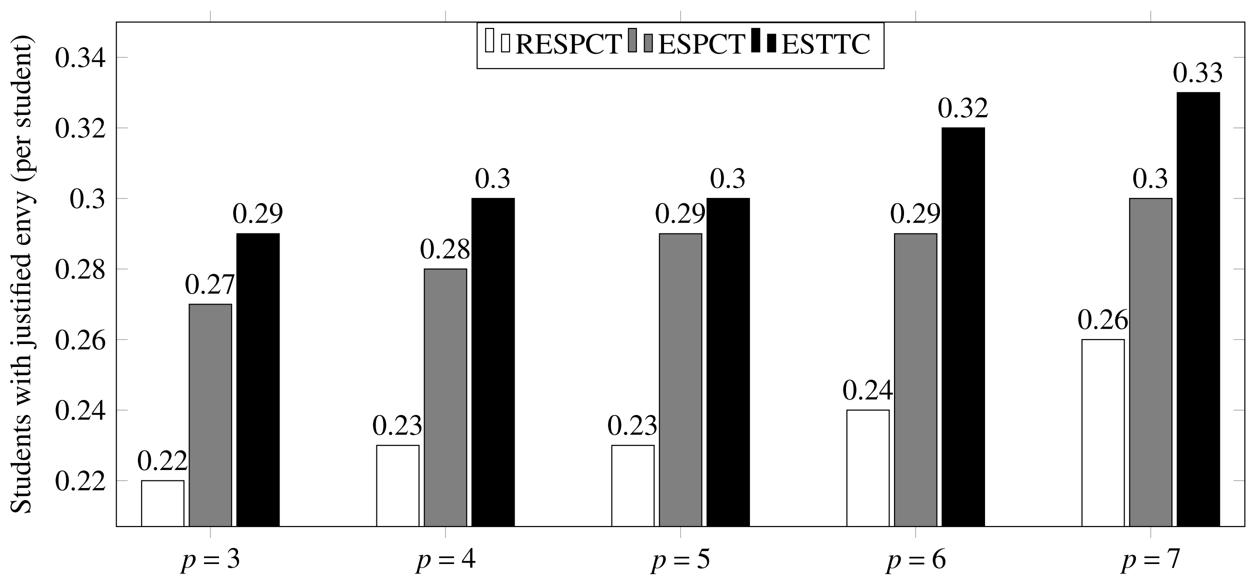


Figure 7 **Students with justified envy (per student)**

While the *number of students with envy* follow the same pattern (see Figure 7), the *number of students envied* (see Figure 8) actually increase across all mechanisms with increasing minimum quotas. However, it can clearly be seen that RESPCT outperforms all other mechanisms in these metrics as well.

6.3. Efficiency

In order to evaluate the gains in efficiency when using RESPCT instead of a strategyproof fair mechanism for matchings with minimum quotas (i.e. one which produces no instances of justified envy), we also implemented the Extended-Seats Deferred Acceptance (ESDA) mechanism by Fragiadakis et al.

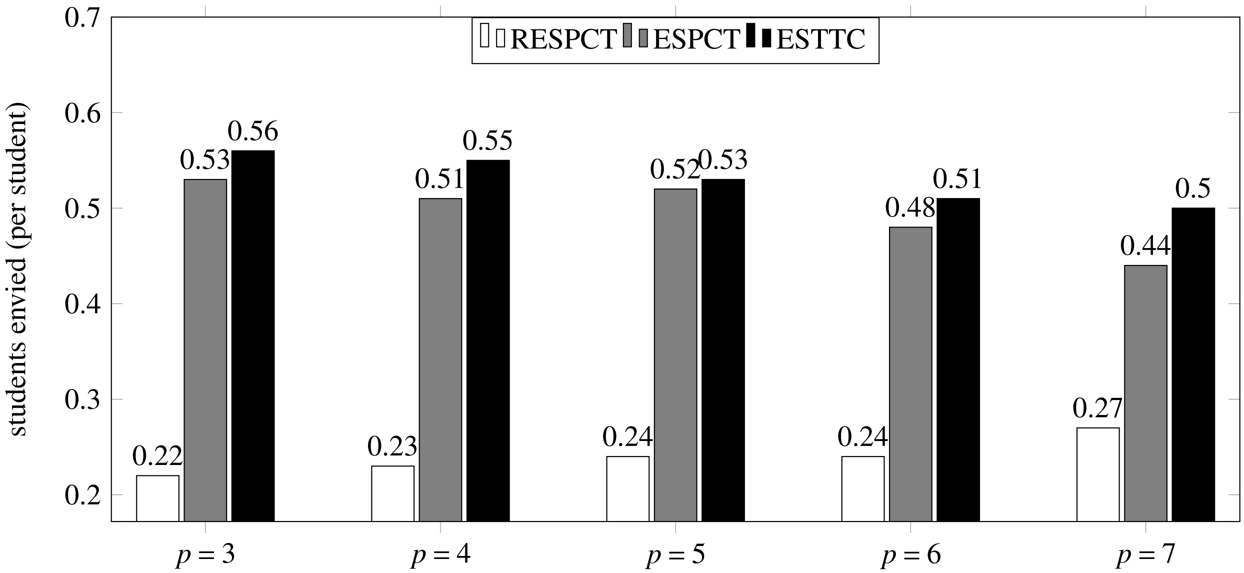


Figure 8 **Students envied (per student)**

	$p=3$		$p=4$		$p=5$		$p=6$		$p=7$	
Rank	RESPCT	ESDA	RESPCT	ESDA	RESPCT	ESDA	RESPCT	ESDA	RESPCT	ESDA
1	2571	2269	2562	2262	2551	2268	2549	2181	2541	2172
2	471	628	487	628	483	629	483	634	479	640
3	223	273	209	276	216	287	209	300	225	291
4	134	180	125	182	127	180	131	189	124	192
5+	284	333	300	335	306	338	311	379	314	388

Table 4 **Comparison of efficiency between RESPCT and ESDA**

(2012). ESDA led to very inefficient outcomes: On average, roughly 10 percent of students could be Pareto-improved given the envy-free matching obtained by ESDA. Figure 9 depicts the average rank of all students, Table 4 shows the distribution of ranks for RESPCT and ESDA. It can be seen, that RESPCT is vastly more efficient based on these measurements, showing once again the advantage of TTC over DA when efficiency is seen as a first-order concern. In Table 5, we report the rank distributions and fairness measures for all mechanisms discussed in this work for the illustrative case of $p = 5$. It can observed once again that the TTC variants are way more efficient than ESDA with RESPECT being the fairest of the efficient mechanisms.

7. Conclusions

In this paper, we introduced a mechanism for the course assignment problem with minimum quotas. We showed that in this environment, ESDA leads to very inefficient assignments. By incorporating a clinching procedure, using a non myopic pointing rule for the courses, and maximizing the number of guaranteed seats at each course, our mechanism, RESPCT, assigned students in a significantly fairer way than ESTTC. RESPCT outperforms ESTTC by far regarding the fairness metrics that we studied. Even

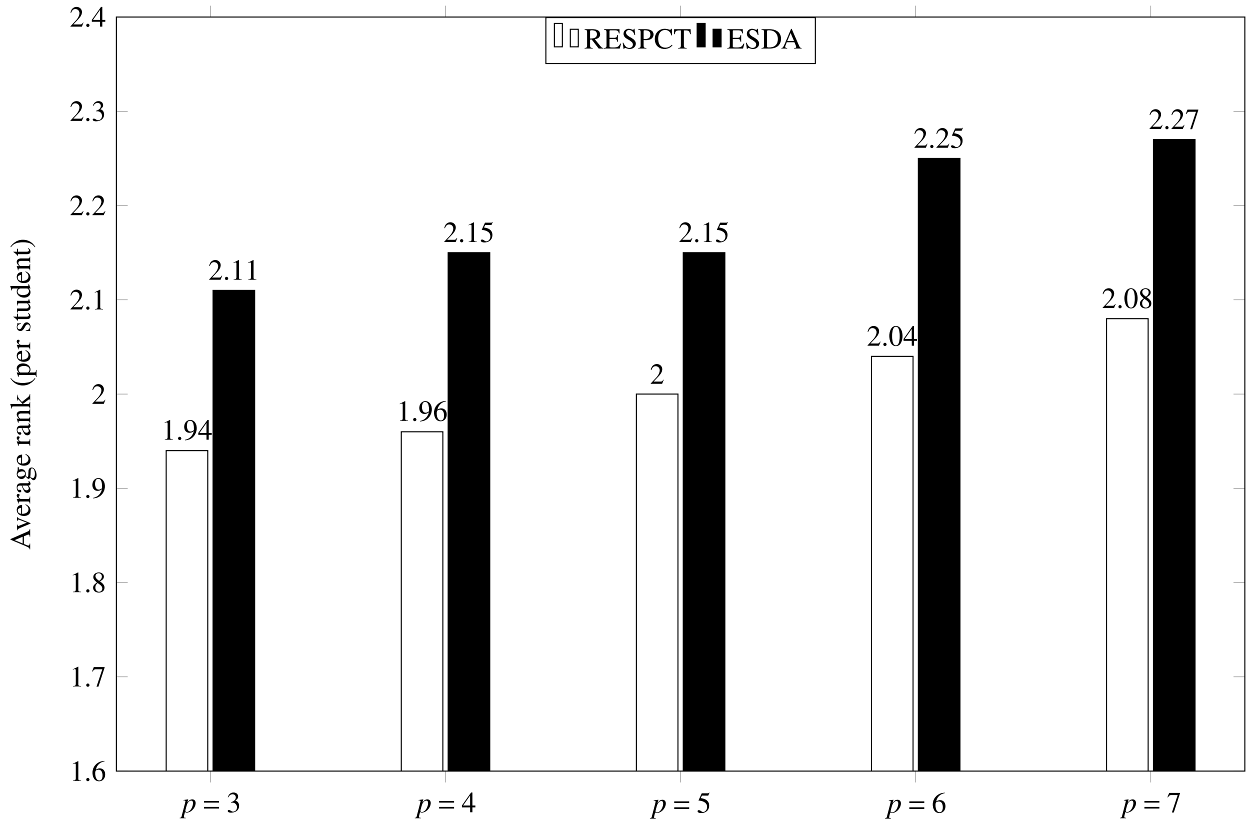


Figure 9 Average rank (per student)

	RESPCT	ESPCT	ESTTC	ESDA
Rank 1	2551	2531	2514	2258
Rank 2	483	467	477	629
Rank 3	216	191	216	278
Rank 4	127	154	138	180
Rank 5+	306	340	338	338
Justified Envy	6.829	22.290	24.503	–
Students with Envy	910	1.089	1.081	–
Students Envied	811	2.107	2.191	–

Table 5 Total number of rank distribution and justified envy for $p = 5$

without widening the range of guaranteed seats, experiments show that only employing clinching and non-myopic pointing, the resulting algorithm, ESPCT, still has significant advantage towards ESTTC, which confirms our expectations considering the fairness behavior of the two mechanisms. The margin at which RESPCT beat the second best performing algorithm ESPCT shrinks as minimum quotas increase and the number of guaranteed seats decreases, yet it stays very high for all values of minimum quotas that we tested.

In our experiments we assume strict preferences, which is a reasonable assumption for many course-assignment applications. In larger applications with many courses students might not provide a complete ranking or be indifferent among a group of courses. Ties lead to various complications (Manlove 1999, Erdil and Ergin 2008, Diebold and Bichler 2017), and we leave this topic for future research.

References

Abdulkadiroğlu, A., P. Pathak, A. Roth. 2009. Strategy-proofness versus efficiency in matching with indifferences: Redesigning the NYC high school match. *The American Economic Review* **88**(5) 1954–1978.

Abdulkadiroğlu, Atila, Yeon-Koo Che, Parag A Pathak, Alvin E Roth, Olivier Tercieux. 2017. Minimizing justified envy in school choice: The design of new orleans' oneapp. Tech. rep., National Bureau of Economic Research.

Abdulkadiroğlu, Atila, Tayfun Sönmez. 2003. School choice: A mechanism design approach. *The American Economic Review* **93**(3) 729–747.

Biró, Péter. 2008. Student admissions in Hungary as Gale and Shapley envisaged. *University of Glasgow Technical Report TR-2008-291* .

Diebold, Franz, Martin Bichler. 2017. Matching with indifferences: A comparison of algorithms in the context of course allocation. *European Journal of Operational Research* **260**(1) 268–282.

Dur, Umut. 2013. A characterization of the top trading cycles mechanism in the school choice problem. *mimeo* .

Ehlers, Lars, Isa E Hafalir, M Bumin Yenmez, Muhammed A Yildirim. 2014. School choice with controlled choice constraints: Hard bounds versus soft bounds. *Journal of Economic Theory* **153** 648–683.

Erdil, Aytek, Haluk Ergin. 2008. What's the matter with tie-breaking? Improving efficiency in school choice. *The American Economic Review* **98**(3) 669–689.

Fragiadakis, Daniel, Atsushi Iwasaki, Peter Troyan, Suguru Ueda, Makoto Yokoo. 2016. Strategyproof matching with minimum quotas. *ACM Transactions on Economics and Computation* **4**(1) 6.

Fragiadakis, Daniel E, Atsuhsi Iwasaki, Peter Troyan, Suguru Ueda, Makoto Yokoo. 2012. Strategyproof matching with minimum quotas. *mimeo (an extended abstract version appeared in AAMAS, pages 1327–1328, 2012)* .

Gale, D., L. S. Shapley. 1962. College Admissions and the Stability of Marriage. *The American Mathematical Monthly* **69**(1) 9–15. doi:10.2307/2312726.

Hakimov, Rustamdjan, Onur Kesten. 2018. The equitable top trading cycles mechanism for school choice. *International Economic Review* .

Kamada, Yuichiro, Fuhito Kojima. 2015. Efficient matching under distributional constraints: Theory and applications. *American Economic Review* **105**(1) 67–99.

Ma, Jinpeng. 1994. Strategy-proofness and the strict core in a market with indivisibilities. *International Journal of Game Theory* **23**(1) 75–83.

Manlove, David F. 1999. Stable marriage with ties and unacceptable partners. *Technical ReportTR-1999-29 University of Glasgow, Department of Computing Science* .

Manlove, David F. 2013. *Algorithmics of Matching Under Preferences*. Series on Theoretical Computer Science, World Scientific Publishing Company Incorporated. URL `http://books.google.de/books?id=7wGJMAEACAAJ`.

Morrill, Thayer. 2013. An alternative characterization of top trading cycles. *Economic Theory* **54**(1) 181–197.

Morrill, Thayer. 2015a. Making just school assignments. *Games and Economic Behavior* **92** 18 – 27.

Morrill, Thayer. 2015b. Two simple variations of top trading cycles. *Economic Theory* **60**(1) 123–140.

Pápai, Szilvia. 2000. Strategyproof assignment by hierarchical exchange. *Econometrica* **68**(6) 1403–1433.

Pycia, Marek, M Utku Ünver. 2017. Incentive compatible allocation and exchange of discrete resources. *Theoretical Economics* **12**(1) 287–329.

Roth, Alvin E. 1982a. The economics of matching: Stability and incentives. *Mathematics of operations research* **7**(4) 617–628.

Roth, Alvin E. 1982b. Incentive compatibility in a market with indivisible goods. *Economics letters* **9**(2) 127–132.

Roth, Alvin E. 2002. The economist as engineer: Game theory, experimentation, and computation as tools for design economics. *Econometrica* **70**(4) 1341–1378.

Roth, Alvin E. 2008. Deferred acceptance algorithms: History, theory, practice, and open questions. *International Journal of Game Theory* **36**(3-4) 537–569.

Shapley, L. S., H. Scarf. 1974a. On cores and indivisibility. *Journal of Mathematical Economics* **1**(1) 23–37.

Shapley, Lloyd, Herbert Scarf. 1974b. On cores and indivisibility. *Journal of Mathematical Economics* **1**(1) 23–37.

Sönmez, Tayfun, Tobias B Switzer. 2013. Matching with (branch-of-choice) contracts at the united states military academy. *Econometrica* **81**(2) 451–488.

Proofs

Theorem 2: PCT is Pareto efficient, strategyproof and satisfies mutual best.

Proof: Since every student assigned receives her favorite among unassigned courses, the mechanism is Pareto efficient. We do not let a student that is pointing at a course clinch that course. Therefore, the clinching process does not effect any existing path. In particular, once there exists a path from a course c to a student s, then that path remains until s is assigned. Therefore, s has no incentive to misrepresent her preferences and point to any course other than her most preferred course. Similarly, a student has no incentive to clinch a course unless it is her most preferred course. She can only clinch a course c if she is guaranteed admissions to c, and once a student is guaranteed admissions to a course, then she never loses that status. This also implies that PCT satisfies mutual best as a student is never assigned a course worse than a course she is guaranteed admissions to. Q.E.D.

Theorem 3 If there are two schools and the total capacity of the two schools is greater than or equal to the number of students, then PCT is fair.

Proof: Label the two schools a and b and let μ be the assignment produced by PCT. After the iterative clinching process, it must be that every student guaranteed a spot at a prefers b to a and *vice versa*. Suppose for contradiction that some student i has justified envy of the assignment μ. Without loss of generality, $\mu(i) = a$ and there exists a j such that $\mu(j) = b$, bP_ia, and $i \succ_b j$. j could not have clinched b or else since i has higher priority and b is i's most preferred school, i would have also clinched b. Therefore, j was assigned via a prioritized trading cycle. Since every student in a cycle is assigned to her most preferred school with available capacity and b must have had available capacity when j was assigned, i could not have been assigned before j is assigned. Note that as there are only two schools, the highest average priority at the schools other than a exactly corresponds to the highest priority at b. Since i has a higher priority at b than j, a does not point at j if i is available. This contradicts j being part of a prioritized trading cycle while i is unassigned.

Theorem 4: ESTTC is strongly group-strategyproof and Pareto efficient and satisfies p-mutual best.

Proof: The proofs for the first two properties are due to Fragiadakis et al. (2012). In the following, we only show p-mutual best: Denote with P_c the set of students who hold one of the p_c highest priorities at c for all $c \in C$. Let $s \in P_c$ be a student whose favorite course is c. Since the minimum capacity of course c in the regular market p_c is equal to the maximum capacity of standard course c in the extended market, s holds one of the q_c highest priorities at the standard course c. Therefore s always points to c. Whenever a student who is not in P_c is assigned c in the extended market, a student in P_c is assigned a course other than c. Thus, no student ever loses its status of holding a priority at c which is among c's remaining maximum capacity and c points to every student s' who is among who holds a priority at c which is among c's remaining maximum capacity at some time if s' is not assigned in an earlier round. Thus, c points to s at some time if s does not receive c in an earlier round due to a trade, i.e. $\mu(s) = c$. Q.E.D.

Theorem 5: Let σ be a vector of guaranteed seats with $\sum_{c \in C} \sigma_c = n$ and $p_c \leq \sigma_c \leq q_c$. Then, (FC) is satisfied.

Proof: Let $C_0 \subset C$. Then,

$$n - |\bigcup_{c \in C_0} G_c| \geq n - \sum_{c \in C_0} \sigma_c = \sum_{c \in C \setminus C_0} \sigma_c \geq \sum_{c \in C \setminus C_0} p_c$$

Q.E.D.

Theorem 6: If $n \geq \sum_{c' \in C \setminus \{c\}} q_{c'} + p_c$ for all $c \in C$, then (FC) is satisfied for $\sigma = q$.

Proof: Let $C_0 \subset C$ and $\gamma \in C \setminus C_0$. Then,

$$\begin{aligned} n - |\bigcup_{c \in C_0} G_c| \geq n - \sum_{c \in C_0} \sigma_c = n - (\sum_{c \in C} q_c - \sum_{c \in C \setminus C_0} q_c) \\ &\geq \sum_{c \in C \setminus \{\gamma\}} q_c + p_\gamma - \sum_{c \in C} q_c + \sum_{c \in C \setminus C_0} q_c \\ &= \sum_{c \in C} q_c - q_\gamma + p_\gamma - \sum_{c \in C} q_c + \sum_{c \in C \setminus (C_0 \cup \{\gamma\})} q_c + q_\gamma \\ &= p_\gamma + \sum_{c \in C \setminus (C_0 \cup \{\gamma\})} q_c \\ &\geq p_\gamma + \sum_{c \in C \setminus (C_0 \cup \{\gamma\})} p_c = \sum_{c \in C \setminus C_0} p_c \end{aligned}$$

Q.E.D.

Theorem 7: Deciding whether a vector σ is incompatible with a minimum quota vector p is a strongly NP-complete problem.

Proof: The problem is obviously in NP, since for a given subset C_0, (FC) can be easily checked. We now proof strong NP-hardness by reduction from vertex cover. Let $G = (V, E)$ be an undirected graph and k an integer. The decision version of vertex cover asks whether there exist k vertices such that all edges E are incident to at least one of these vertices. For G and k we construct the following course assignment instance:
For each vertex $v \in V$ create one course c_v with minimum quota $p_{c_v} = 1$ and maximum capacity $q_{c_v} = |E|$.
For each edge $e \in E$ create a student s_e.
Additionally, create $\alpha = |V| - k - 1$ auxiliary students $d_1, \ldots, d_\alpha$.
Define priorities, guarantees and σ of courses such that each course c_v guarantees seats to all students s_e for which e is incident to v in G. Auxiliary students are guaranteed a seat at no course.

Then, there exists a vertex cover in G of size k if and only if σ is not compatible with p.

Let $v_1, \ldots, v_k$ be a vertex cover. Since each edge is incident to at least one of these vertices, all students representing these edges have guarantees at courses $c_{v_1}, \ldots, c_{v_k}$. Thus for $C_0 = \{c_{v_1}, \ldots, c_{v_k}\}$, it holds that

$$|E| + \alpha - |\bigcup_{c \in C_0} G_c| = \alpha < |V| - k = \sum_{c \in C \setminus C_0} p_c$$

, leading to an incompatibility of σ and p.
Since α students are not guaranteed a seat at any courses and all courses have a minimum quota of 1, there must exist a subset of courses C_0 of size at most k, such that $|\bigcup_{c \in C_0} G_c| = |E|$. In this case, all edges representing those students have to be incident with at least one vertex representing a course in C_0. Thus, these vertices form a vertex cover for G.
Q.E.D.

Theorem 8: RESPCT satisfies strategyproofness, Pareto efficiency and σ-mutual best such that $\sum_{c \in C} \sigma_c \geq n$.

Proof: Every student and course is part of at most one cycle during each round of modified ESPCT. Further, no student can receive a course by clinch that she could have otherwise not received: Denote G_c^k the set of students who are guaranteed a seat at course c in step k. Assume student s ∈ S can receive a seat at regular course c ∈ C during the clinching phase of some step of RESPCT. Thus, s ∈ G_c^k during some round k. In RESPCT, since $|G_c| = \sigma_c$, c and c^*

combined point to at least $|G_c|$ students who are in G_c until σ_c expires. If a student who is not in G_c^k is assigned either c or c^*, a student in G_c^k must be assigned a course other than c and c^*. Hence, either c or c^* point to every student $s' \in G_c^k$ for all rounds k unless s' receives c or c^* by either clinch or by trading an endowment from a different course. Consequently, s could also have received c during the trading phase.

Strategyproofness: No student can improve her assignment by pointing to a course other than her most preferred one. Assume that student s can profit by misrepresenting her preferences by pointing to course d instead of her favorite course c and let k be the last round where she can profit from that misrepresentation. We differentiate the following two cases: **Case 1**: d is a standard course. Again, this case is equivalent to part three of Theorem 4. **Case 2**: d is an extended course. If s does not build a cycle with d in round k, she does not profit from misrepresenting, which is a contradiction to our assumption. Thus, s must build a cycle with d. Denote Z^{*k} the set of students who are endowed a seat at an extended course at the beginning of round k and $Z^k(s,d)$ the set of students who are involved in the cycle between s and d and are endowed a seat at an extended course at the beginning of round k. Since the student who is endowed d is involved in this cycle, $|Z^k(s,d)| \geq 1$. Thus, no course c^* which points to a student $s' \in Z(s,d)$ is assigned in round k if s points to c. Since all extended courses point to at most ϵ^k students in round k, $Z^{*k} \leq \epsilon^k$. If s pointed to c instead of d in round k, no cycle could have existed in round k between d and any other student and consequently, no student in $Z(s,d)$ is assigned a course in round k. Therefore, $\epsilon^{k+1} \geq \epsilon^k - |Z^{*k}| + |Z^k(s,d)| \geq \epsilon^k - \epsilon^k + 1 \geq 1$. Therefore, the extended courses are not taken off the market before the beginning of round $k+1$ and s can still build a cycle with d in round $k+1$. Thus, k is not the last round in which s can profit from misrepresenting, which is a contradiction to our assumption.

As described above, no student can clinch at a course where she could not have received a seat during the cycle resolution phase. Therefore, no student can make herself better off by clinching some course d different from her most preferred one c since she could have also received d during the clinching phase. Since no student can make herself better off by lying during the cycle resolution phase, it is obvious that a student can neither improve her assignment by lying during the clinching phase nor by lying during the cycle resolution phase. Consequently, RESPCT is strategyproof.

Pareto efficiency: The proof is equivalent to part two of proof of Theorem 4.

σ-mutual best: Any vector σ with $\sum_{c \in C} \sigma_c = n$ satisfies FC due to Theorem 5, thus any method described in Section 5.1 produces such a vector σ. Then, if a student has one of the σ_c highest priorities at her most preferred course c, she clinches c during the clinching phase of step 2 of RESPCT and, therefore, RESPCT satisfies σ-mutual best. Q.E.D.