

How to Assign Scarce Resources Without Money: Designing Information Systems that are Efficient, Truthful, and (Pretty) Fair

Martin Bichler, Alexander Hammerl, Stefan Waldherr

Department of Informatics, Technical University of Munich, Germany, {bichler, waldherr}@in.tum.de

Thayer Morrill

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

Matching with preferences has great potential to coordinate the efficient allocation of scarce resources in organizations when monetary transfers are not available, and thus can provide a powerful design principle for information systems. Unfortunately, it is well-known that it is impossible to combine all three properties of truthfulness, efficiency, and fairness (i.e. envy-freeness) in matching with preferences. Established mechanisms are either efficient or envy-free, and the efficiency loss in envy-free mechanisms is substantial. We focus on a widespread representative of a matching problem: course assignment where students have preferences for courses and organizers have priorities over students. An important feature in course assignment is that a course has both a maximum capacity and a minimum required quota. This is also a requirement in many other matching applications such as school choice, hospital-residents matching, or the assignment of workers to jobs. We introduce RESPCT, a mechanism that respects minimum quotas and is truthful, efficient, and has low levels of envy. The reduction in envy is significant and is due to two remarkably effective heuristics. We follow a design science approach and provide analytical and experimental results based on field data from a large-scale course assignment application. These results have led to a policy change and the proposed assignment system is now being used to match hundreds of students every semester.

Key words: Top Trading Cycles, Course Assignment, Design Science

Version June 14, 2020

1. Introduction

Matching theory began with the problem of how to match two distinct sets of agents (Gale and Shapley 1962). Gale and Shapley's celebrated marriage problem asks the question of how to assign men and women into a stable set of marriages. The distinguishing features of this model are that there are two distinct sets of agents and money cannot be used to aid in the assignment. The central question of the marriage problem is the existence and properties of an equilibrium assignment; in this context, it is called a stable assignment.¹

¹ An assignment is individually rational if no man or woman wants to divorce his or her partner. An assignment has a blocking pair if there is a man and a woman who prefer each other to their current assignment. An assignment is stable if it is individually rational and there is no blocking pair.

There is no tension between stability and efficiency here; since a stable assignment is in the core, it is Pareto efficient.

A closely related question is how to match an agent to an object. An important example of this is the school assignment problem (Balinski and Sönmez 1999, Abdulkadiroğlu and Sönmez 2003). The critical distinction between these two models is that in school assignment, the schools are viewed as “objects to be consumed” and are not included in welfare considerations. Schools are not interpreted as having preferences over students; instead, a student’s priority at a school is interpreted as her “right” to attend the school. The central objective of a school assignment is not finding an equilibrium, since students are not allowed to change schools unless they are granted permission. Instead, the central question is what constitutes a *fair* assignment, in the sense of respecting each student’s rights at the schools. Formally, student i has justified envy of student j if i prefers j ’s school to her own and has higher priority than j at that school. An assignment is interpreted as fair if no student has justified envy of another.

Our focus is on course assignment within educational institutions, a widespread application of one-to-many object assignment (without transfers). This problem is analogous to the school assignment problem, as it is a one-to-many assignment problem where the objects have priorities over the students.² Depending on the application, course assignment could be a many-to-many (if a student is assigned several courses) or one-to-many (if a student is assigned to only one course out of many) problem. One-to-many assignments (similar to school choice problems) are ubiquitous at European universities. For example, in larger European universities, there is a study plan with fixed lectures at particular times of the week that students need to attend, but the students can choose different seminars, tutorials for the lectures, or lab courses. In our specific application, the course instructors rank the students; however, in general, one student can have higher priority at a course for many reasons, such as seniority, honors status, prior experience with a subject, or his motivation letter. In a particular semester, students only have one choice out of many seminars (or lab courses). This is also the case at the Technical University of Munich. Course organizers offer seminars or lab courses on specific topics (robotics, programming, compilers, machine learning, etc.) and have priorities for students. For example, for a robotics course with 15 seats, a course organizer might prefer computer science students in the master’s program over students in industrial engineering or bachelor’s students in general. Moreover, course organizers can (and do) submit individual rankings over students based on achievements or grades from previous courses. Depending on the focus of these courses, their priorities will differ.

Although course assignment is a specific application, it is very similar to other applications, such as school choice (Abdulkadiroğlu and Sönmez 2003), college admission (Biró 2008), cadets-to-branch

² This course assignment problem is different from the course scheduling problems described by Budish et al. (2016), where students need to select packages of course seats in different courses, which is also referred to as the combinatorial assignment problem.

matching in the military (Sönmez and Switzer 2013), the well-known hospital-residents matching (Roth 2002, NRMP 2014), refugee resettlement (Delacrétaz et al. 2016), or various labor markets (Biró 2017). A key difference between our problem and these is that courses typically require a minimum number of students. While this is relevant in other applications, it has not been central to the prior literature on school choice or hospital-residents matching. Still, the findings in this study are of interest to many other application domains of matching with preferences.

Efficiency, fairness, and incentive-compatibility (aka truthfulness) are seen as the key design desiderata for matching problems with preferences, and they are also central to course assignment. A matching is Pareto *efficient* if it is impossible to reallocate so as to make any one student strictly better off without making at least one student worse off. A matching is *fair* or *envy-free* if no pairs of students and course organizers would like to switch partners, as introduced earlier. Finally, *strategyproofness* of a matching mechanism refers to when students have a dominant strategy for reporting their true preferences to the mechanism. This is a strong notion of *incentive-compatibility*.³ The central tension in course assignment, school choice, or any object assignment where the objects have priorities over the agents is that there does not always exist a fair assignment that is Pareto efficient (Abdulkadiroğlu and Sönmez 2003).

Our paper has multiple connections to information systems literature. In a widely cited article by Ba et al. (2001), the authors argue that incentives and incentive-alignment need to be considered in information systems design. Various subsequent papers emphasized incentives in the design of information systems for supply chain coordination (Fan et al. 2003) and electronic markets (Bapna et al. 2003, Adomavicius and Gupta 2005, Bapna et al. 2009, Liu et al. 2010, Adomavicius et al. 2013). Some of these authors use lab experiments to study new computational market designs (Adomavicius et al. 2013, Bichler et al. 2017), while others analyze new market designs in the field as we do (Lu et al. 2017, 2016). Bichler et al. (2010) provide a broader perspective on market design in information systems that combines economic principles with information systems design. Almost all of this literature is focused on auction-based mechanisms, however. Our paper fits very well into this overall line of research, but adds a new perspective: the designs we introduce do not require monetary transfers to achieve an efficient allocation of resources, as is required in auctions. This opens up a variety of new applications within and across organizations. Often tasks must be assigned to workers, operating rooms to surgeons, or course seats to students, but payments cannot be used and preferences cannot be easily expressed as comparable cardinal numbers.⁴

In particular, our work is a design science contribution to matching with preferences. Design science supports a pragmatic research paradigm that calls for the creation of innovative artifacts to solve real-world

³ Note that course organizers are assumed to be non-strategic and have publicly known priorities among groups of students. Matching mechanisms that are incentive-compatible for both sides of a matching market are impossible.

⁴ Note that for assignment problems where agents can state cardinal preferences and it is possible to exchange money among participants, strategyproof, efficient, and fair mechanisms do exist (Sun and Yang 2003).

problems (Hevner and Chatterjee 2010). Let us briefly address the main elements of the guidelines outlined in Gregor and Hevner (2013).

- **Purpose:** We make a design science contribution (Hevner et al. 2004) and introduce a strategyproof and fully efficient mechanism that exhibits surprisingly little envy. The approach considers minimum quotas, which adds to the complexity of the problem.
- **Scope and relevance:** The paper addresses a fundamental assignment problem with preferences, which is widespread in the form of course assignment at universities. Similar matching problems appear in many other domains.
- **Method:** We first propose an algorithmic solution and formally prove important properties.
- **Artifact and evaluation:** We then implement a respective artifact and use field data from a large-scale application where up to 7,000 students are matched every year. The data illustrates the benefits for the specific application with respect to reduced envy and provides the basis for the adoption of our approach at the Technical University of Munich.

In summary, our paper provides an artifact to solve a relevant real-world problem that follows the seven design science guidelines from Hevner et al. (2004) and addresses the relevance, rigor, and design cycles outlined in Hevner (2007).

The structure of the paper follows the publication schema by Gregor and Hevner (2013). In Section 2, we discuss course assignment in general and the related literature, demonstrate the previous solution for assigning courses, and highlight the contributions of our new mechanisms relative to prior literature. Afterwards, we formally define the course assignment model in Section 3. Our method is algorithmic design and we use formal proofs to characterize the properties of the overall artifact. In Section 4, we discuss our new artifact in detail, laying out all contributions that lead to a significant increase in fairness. Then, we provide results based on field data in Section 5. Finally, we discuss and interpret our results in Section 6 and draw conclusions in Section 7.

2. Literature Review

In what follows, we first explain Gale’s Top Trading Cycles algorithm and show why there is tension between efficiency and fairness. Next, we describe the options a designer has to reduce envy while still respecting requirements such as minimum quotas that are relevant in many assignment problems with preferences.

2.1. Top Trading Cycles

A market designer typically can decide between two strategyproof mechanisms: Gale’s Top Trading Cycles algorithm (hereafter TTC), which is strategyproof and efficient but not envy-free, and Gale and Shapley’s Deferred Acceptance algorithm (hereafter DA), which is strategyproof and envy-free but not efficient.

In their seminal paper, Shapley and Scarf (1974) introduced TTC in order to find a competitive solution for a stylized housing market. In the Shapley-Scarf model, each agent is endowed with an object (a house). A round of TTC proceeds as follows. Each agent points at her favorite remaining object, and each object points to its owner. There must exist at least one cycle. For each cycle, assign the agent to the object she is pointing to and then remove the agent and object. The mechanism proceeds by repeating this process until no agents or objects remain.

EXAMPLE 1. To illustrate the TTC algorithm, consider the preference profile of four agents $\{1, 2, 3, 4\}$ for the objects $\{a, b, c, d\}$ in Table 1.⁵ Initially, agent 1 owns object a , agent 2 owns object b , agent 3 owns object c , and agent 4 owns object d .

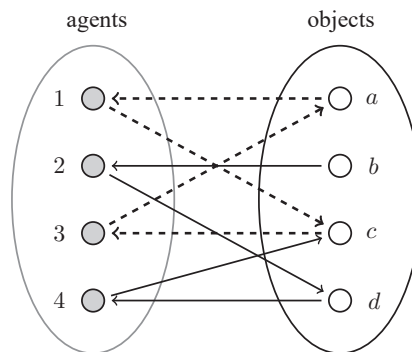


Figure 1 Illustration of the first round of the TTC algorithm as a bipartite graph for the example in Table 1.

\succ_1	\succ_2	\succ_3	\succ_4
c	d	a	c
b	a	d	b
d	b	c	a
a	c	b	d

Table 1 Preferences of agents in TTC

The first round of the TTC algorithm with these preferences is illustrated in Figure 1. The dashed lines mark a cycle. Object c is assigned to agent 1, and object a is assigned to agent 3. These objects and agents are then removed. In the second round, there remains a bipartite graph with the nodes 2, 4, b , and d . Agent 2

⁵ For the formal definitions we refer to Section 3. For each agent, a column represents her preference order over objects, with an object a being preferred over object b if a is in a row above b 's row.

Table 2 (a) Course priorities (\succ_c) and (b) student preferences (\succ_s)

(a)	\succ_{c_1}	\succ_{c_2}	\succ_{c_3}	(b)	\succ_{s_1}	\succ_{s_2}	\succ_{s_3}
	s_1	s_2	s_2		c_2	c_1	c_1
	s_3	s_1	s_1		c_1	c_2	c_2
	s_2	s_3	s_3		c_3	c_3	c_3

points to object d and agent 4 to object b . These nodes form a new cycle and are assigned accordingly. The final matching is $\{(1, c), (2, d), (3, a), (4, b)\}$.

Roth (1982) 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) was 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 include Abdulkadiroğlu et al. (2017), Morrill (2013), Morrill (2015a), and Dur (2013).

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.

2.2. Efficiency vs. Envy-Freeness

A central problem in matching with preferences is the efficiency/envy-freeness tradeoff discussed by Abdulkadiroğlu and Sönmez (2003) in their seminal paper. Let us introduce a brief example to illustrate this tradeoff:

EXAMPLE 2. Consider the course allocation problem with three students $S = \{s_1, s_2, s_3\}$ and three courses $C = \{c_1, c_2, c_3\}$, each with a capacity of 1. The course priorities (\succ_c) and the student preferences (\succ_s) are given in Table 2 as strict ordinal rankings.

Let’s take a look at two possible matchings, μ_1 and μ_2 , and their consequences from the students’ perspective. The matching

$$\mu_1 = \begin{pmatrix} c_1 & c_2 & c_3 \\ s_1 & s_2 & s_3 \end{pmatrix}$$

is the only fair matching; it is underlined in Table 2. However, the matching is Pareto dominated for the students by

$$\mu_2 = \begin{pmatrix} c_1 & c_2 & c_3 \\ s_2 & s_1 & s_3 \end{pmatrix}.$$

Fairness forces students s_1 and s_2 to share the courses in an inefficient way. If students s_1 and s_2 were assigned the courses c_2 and c_1 , respectively, then student s_3 would prefer course c_1 to her assignment c_3 , and she would have higher priority for course c_1 than its assigned student, s_2 . In this case, s_3 would have justified envy of s_2 . *As a result, fairness conflicts with Pareto efficiency!*

The recommendation by Abdulkadiroğlu and Sönmez (2003) regarding DA is clear: DA Pareto dominates any other envy-free assignment. However, their recommendation regarding TTC is far less certain. Surprisingly little is known about the degree of envy in TTC or alternative efficient mechanisms when multiple units of an object are assigned to more than one agent, e.g. when many students are assigned to the same course. We show that the inefficiency of DA is substantial in course matching and that TTC generates significant envy. Therefore, a natural question is: How we can make an efficient mechanism with as little envy as possible?

2.3. Reducing Envy and Considering Quotas

Let us now introduce possibilities for the reduction of justified envy in TTC. 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 or quotas, a natural analog to individual rationality is respecting top priorities: if a course or object a has capacity for q agents, then each of the q highest ranked agents at a are either assigned to a or a course 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 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 courses other than a . A designer cannot know which courses an agent will point to (and the mechanism 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.

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 and 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 course. The authors proposed, among others, the Extended-Seat DA and Extended-Seat TTC algorithms. While the former is group strategyproof⁶ and envy-free, the latter is strongly group strategyproof and Pareto efficient. However, the Extended-Seat TTC does not consider the level of envy of the resulting matching, and this can be high as we show. Unfortunately, with Extended-Seat TTC it is much harder to determine which students are guaranteed a course and, consequently, *reducing justified envy* becomes a very challenging problem, one that we address in our paper.

As far as we know, Abdulkadiroğlu et al. (2017) is the only other paper that compares the performance of TTC to the 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 to alternative implementations and find an essentially negligible impact on the envy of the assignment. School assignment and course assignment are closely related problems, but there are also substantial differences. In our field data from the course assignment application, we find far more heterogeneity in preferences than in these reports on the school choice problem. 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. The only paper related to ours in the context of matching is Hakimov and Kesten (2018), which also asks whether it is possible to reduce the envy of TTC. This approach does not allow for minimum quotas, which are of central importance in most course assignment problems. Incorporating minimum quotas into ETTC is non-trivial. Doing so would constitute another new algorithm (which maybe uses some of RESPCT’s ideas). A few papers deal with more complex matching problems, such as the multi-unit or combinatorial assignment of course schedules (Budish et al. 2016). Students in MBA programs can often select course schedules and not just one out of many seminars. This leads to a different problem, and we do not consider such course assignment problems in our paper.

2.4. The Prior Solution in Practice

Our specific task was to assign students to classes in the computer science and information systems department at the Technical University of Munich, a large European university. This department is currently the largest department at the university with more than 6,000 students. Students have to enroll in one out of many seminars in one semester and one out of many practical courses in another semester, and the number

⁶ Group strategyproofness means that no group of participants can collude and misreport their preferences in a way that makes every participant better off.

of assignments 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 one out of many tutor groups. These course-specific assignments add up to another $\sim 7,000$ assignments of students to courses every year in the computer science department alone. Based on its success, the matching system was also adopted by other departments, including architecture, business, mathematics, and mechanical engineering.

Previously, students have 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 et al. 2014, Diebold and Bichler 2017). In FCFS, a student would often sign up for more than a dozen of courses, but then cancel the registrations for all but the most preferred one among those courses where she was admitted. Due to the almost random order of arrivals, simple FCFS rules are neither efficient nor fair. For example, students registered when their most preferred courses were already taken by students who had a lower priority for the course organizer. Students tried to swap with others who had a seat in a course higher on their preference list. At the same time, course organizers did not get the students who were high on their priority list. The computer science department 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, and overall was viewed as a great improvement over FCFS. However, *DA does not allow for minimum quotas*, which is central in most course assignment problems, and *DA incurs significant efficiency losses*, as we will also show later. 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. In large-scale applications with hundreds of participants, envy is less of a concern to students than in small applications with only a dozen participants. On the other hand, efficient use of scarce resources is a key concern to the Dean of Studies. Moreover, preferences in course assignments are very different from assignments 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 nearly Pareto efficient (see for example Abdulkadiroğlu et al. (2017), which describes such assignments in both Boston and New Orleans). However, as we will show, this finding does not generalize and DA is quite inefficient in our course assignment environment where preferences are more heterogeneous. An initial analysis showed that the efficiency loss in DA is substantial compared to TTC.

Second, a number of core courses require a minimum number or *minimum quota* of attendees. Fragiadakis et al. (2016) mention computer science students at Kyushu University who must all complete a laboratory

requirement. This characteristic is typical for course assignment applications in general. In our situation, courses in profile areas shall be offered by the department in any case and a minimum number of students is required by the Dean of Studies. This seemingly innocuous feature can be decisive for the adoption of a matching mechanism for course assignment, but it leads to substantial challenges in the implementation, as we show in this paper.

2.5. A New Approach

We will demonstrate that we can assign agents to objects (i.e., students to courses) with significantly less envy than TTC (or Extended-Seat TTC), which makes our mechanism very appealing as a practical alternative to DA and TTC. Our proposed assignment mechanism is called **Extended Seat Prioritized Clinch and Trade** with a widened **Range** of guarantees (hereafter **RESPCT**). We explicitly consider minimum quotas in RESPCT, which turned out to be a central requirement for our course allocation problem, as well as for a wide range of applications beyond the scope of this paper. For example, schools require a minimum number of students, a military branch requires a minimum number of cadets, and rural hospitals require a minimum of doctors to operate. Often, minimum quotas are implemented as a means to guarantee diversification of quality or maintain racial and ethnical balance (Abdulkadiroğlu and Sönmez 2003, Ehlers et al. 2014).

We theoretically characterize the important properties of the RESPCT mechanism regarding strategyproofness, efficiency, and fairness. For many assignment problems, there is no data publicly available. We are fortunate to be able to compare the performance of our algorithm in the context of a large-scale course assignment application from the Technical University of Munich. In all, we utilized 10 different assignments ranging from 27 to 733 different students. We compared instances of the algorithm for a variety of minimum quota scenarios. In every scenario, our algorithm had substantially less envy than the alternatives, resulting in a reduction of instances of justified envy by a factor of three or more. We also found substantial improvement regarding the number of students that envy other students. In summary, our assignment is efficient and has far fewer students with justified envy compared to TTC or quota-respecting versions of TTC. Compared to (quota-respecting versions of) DA, roughly 10 percent of the students could be Pareto-improved with RESPCT. These significant improvements were the result of two algorithmic innovations: prioritized pointing and maximizing the seat guarantees. It is the combination of these algorithmic contributions that led to a remarkable reduction in justified envy.

Strategyproofness, efficiency, and very low levels of envy convinced the computer science department at the university to change the assignment procedure from DA to RESPCT. Note that our results are relevant beyond course assignment. There has been an intense discussion about efficiency and fairness in school choice (Abdulkadiroğlu and Sönmez 2003), but the result also matters for the assignment of medical residents (Kamada and Kojima 2015), college admissions (Biró 2008), and other matching problems that

are frequently found in organizations. If there are only a few cases of justified envy, a market designer might aim for efficiency rather than fairness as first-order priority in all of these applications.

3. Model and Method

Before we formally introduce important design desiderata, let us first introduce basic notation and the TTC algorithm.

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, $S = \{s_1, s_2, \dots, s_n\}$ is the set of students, while $C = \{c_1, c_2, \dots, c_m\}$ is the set of courses. The vector $p = (p_1, p_2, \dots, p_m)$ describes the courses' minimum quotas, while $q = (q_1, q_2, \dots, q_m)$ is the vector of their maximum quotas (capacity). We assume that $p_c \leq q_c$ for all c and $\sum_{c \in C} p_c \leq n \leq \sum_{c \in C} q_c$.

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 . Let \mathcal{P} be the set of all strict preference orders a student can have. Then, $\succ_s \in \mathcal{P}^{|S|}$ describes the preference profiles for all students S . As is common in this literature, we assume that preferences are independent and private and do not change when students observe the preferences of others. Correspondingly, each course $c \in C$ has a strict, complete, irreflexive, and transitive preference relation \succ_c over students. A matching $\mu : C \cup S \mapsto 2^S \cup C$ is a function that 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 $s \in S$, $c \in C$.

The one-to-many generalization of TTC introduced by Abdulkadiroğlu and Sönmez (2003) is represented by Algorithm 1.

Algorithm 1 Top Trading Cycle

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

Round $k \geq 1$. 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.

The 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. We obtain $\tilde{>}_s$ by inserting every extended course c^* in every student's preference profile directly behind the corresponding standard course. Thus, if $>_s = c_i >_s c_j >_s \dots$ for some $s \in S$, $\tilde{>}_s = c_i \tilde{>}_s c_i^* \tilde{>}_s c_j \tilde{>}_s c_j^* \dots$ in the extended market. Let $\epsilon = n - \sum_{c \in C} p_c$ be the number of students who can at maximum be assigned to extended courses without making a feasible matching impossible. In addition to the individual preferences of courses, there also exists an additional global preference list $>_{ML}$ (master 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.

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 its favorite student, and each extended course points to the student with the highest priority according to $>_{ML}$. 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 any is reduced to zero, it is also removed. If the number of students assigned to extended courses reaches ϵ after a round, remove all extended courses.

Denote by $\tilde{\mu}$ the matching in the extended market. Then, the matching μ in the regular market is defined as follows: $s \in \mu(c)$ 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 until the number of students assigned to extended courses reaches ϵ , at which point all extended courses are taken off the market. Therefore, ESTTC always results in a matching that 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 improved to make the mechanism fairer, as described in Section 4.

EXAMPLE 3. Let $S = \{s_1, \dots, s_6\}$ and $C = \{c_1, c_2, c_3\}$. The preferences, priorities, and quotas are provided in Table 3. 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. Therefore the number of students assigned to extended courses 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 to 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

\succ_{s_1}	\succ_{s_2}	\succ_{s_3}	\succ_{s_4}	\succ_{s_5}	\succ_{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

\succ_{c_1}	\succ_{c_2}	\succ_{c_3}	\succ_{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 3 Preferences and priorities for student-to-course matching with minimum quotas.

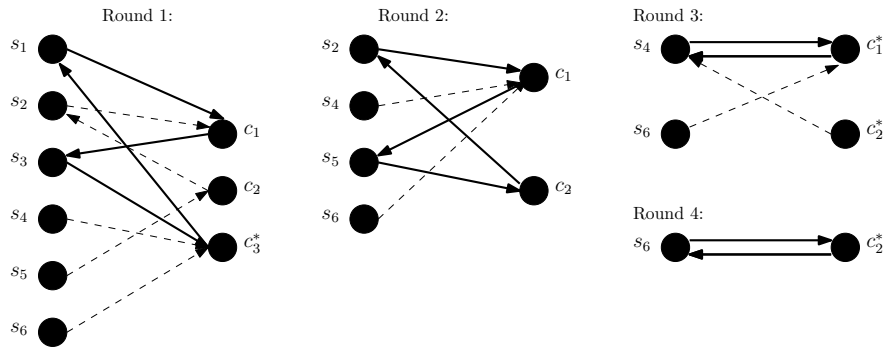


Figure 2 Run of ESTTC for Example 3

capacity reaches 0.

Round 2: We obtain c_1, s_5, c_2, s_2, c_1 as the 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^* . We then 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.

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 $\chi : \mathcal{P}^{|\mathcal{S}|} \mapsto \mathcal{M}$ is a function that 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. We define \succeq_s by $a \succeq_s b$ if and only if $a \succ_s b$ or $a = b$.

DEFINITION 1 (STRATEGYPROOFNESS). A mechanism χ is strategyproof if and only if for any $\succ_s \in \mathcal{P}^{|\mathcal{S}|}$, $\zeta \in \mathcal{S}$ and $\succ'_\zeta \in \mathcal{P}$, $\chi_{\succ_s}(\succ_s) \succeq_\zeta \chi_\zeta(\succ_{s \setminus \{\zeta\}}, \succ'_\zeta)$, where $\chi_\zeta(\succ_s)$ is the matching that ζ receives under \succ_s .

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

DEFINITION 2 (PARETO EFFICIENCY). A matching μ' *Pareto dominates* another matching μ if there exists some $\zeta \in S$ such that $\mu'(\zeta) \succ_{\zeta} \mu(\zeta)$ and for all other $\zeta \in S$ it holds that $\mu'(\zeta) \succeq_{\zeta} \mu(\zeta)$. A matching μ is Pareto efficient if and only if no other matching Pareto dominates μ .

DEFINITION 3 (FAIRNESS / ENVY-FREENESS). A matching μ is envy-free if and only if $\mu(\zeta') \succ_{\zeta} \mu(\zeta)$ implies $\zeta' \succ_{\mu(\zeta')} \zeta$ for all $\zeta, \zeta' \in S$. If $\mu(\zeta') \succ_{\zeta} \mu(\zeta)$ and $\zeta \succ_{\mu(\zeta')} \zeta'$ for some ζ' , we say that ζ and c form a blocking pair or ζ has *justified envy* toward ζ' .

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 (see example 2). Example 3 contains four instances of justified envy: s_6 has justified envy toward s_2 , s_3 , and s_4 and s_4 has justified envy toward s_3 . Note that envy-free matchings may not always exist with quotas. If all student-course pairs are acceptable (every student prefers any course to being unmatched, and any course organizer prefers any student to a vacant seat), then an envy-free matching always exists. If not all pairs are acceptable, then an envy-free matching might not exist.

As we discussed earlier, the DA mechanism by Gale and Shapley (1962) is strategyproof and fair while TTC is strategyproof and Pareto efficient (Shapley and Scarf 1974). 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. Since fairness is not achievable for TTC even without minimum quotas, we relax fairness in the following definition. A weaker fairness condition is *mutual best*:

DEFINITION 4 (MUTUAL BEST). A matching μ satisfies mutual best if and only if the following condition is met: c is the favorite course of s and s is among the q_c highest priorities of c , which 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*.

If a student s is guaranteed a seat at course c , s will never cause justified envy. She has no justified envy toward any student since she ends up at her favorite course. Also, no other student s' can have justified envy toward s , since in order to have justified envy, s' needs to have a higher priority at c . However, in this case s' also is guaranteed a seat at c and would either end up at c or a course s' prefers over c .

TTC satisfies mutual best; if s is guaranteed a seat 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, using a serial dictatorship or linear programming procedure to maximize the size of the matching do not satisfy mutual best (Morrill 2013). Unfortunately, when minimum quotas are a requirement, no mechanism always satisfies mutual best.

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

\succ_{s_1}	\succ_{s_2}	\succ_{c_1}	\succ_{c_2}
c_1	c_1	s_1	s_2
c_2	c_2	s_2	s_1
		p	0
		q	2
			1
			2

Table 4 Example of a mutual best violation

Proof: Consider an example with $S = \{s_1, s_2\}$ and $C = \{c_1, c_2\}$ and preferences as in Table 4.

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 met: (c is the favorite course of s and s is among the σ_c highest priorities of c) implies that $\mu(c) = s$.

Note that 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. In this paper, we will discuss how to guarantee as many seats as possible, i.e. how to maximize σ . 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. In the next section, we will introduce the RESPCT mechanism and formally prove its important properties.

4. Artifact Description

As we will show, the RESPCT mechanism includes a number of innovations that improve the fairness of TTC. We first discuss improvements without minimum quotas and then consider these quotas. Our approach of making TTC fairer leverages two observations regarding the one-to-many variant of the assignment problem. Envy in TTC can be caused by unnecessary trades and allowing early trade opportunities for students that have low priorities at other courses. The former can be mitigated by a clinching procedure that we discuss in Section 4.1, the latter by prioritized pointing, which we introduce in Section 4.2. Afterwards, we show in Section 4.3 that these innovations can also improve fairness when considering minimum quotas, while also retaining strategyproofness and Pareto efficiency. Since the performance of both clinching as well as prioritized pointing relies heavily on the guarantees that courses can offer, we show how to maximize these number of guaranteed seats in Section 4.4. Finally, we describe the complete RESPCT mechanism in Section 4.5.

4.1. Clinching

Morrill (2015b) already discussed that there are unnecessary trades in TTC that cause unfairness without contributing to TTCs Pareto efficiency, strategyproofness, and mutual best properties. Consider the following example:

EXAMPLE 4. 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:

\succ_{s_1}	\succ_{s_2}	\succ_{s_3}	\succ_{c_1}	\succ_{c_2}
c_2	c_1	c_2	s_1	s_2
c_1	c_2	c_1	s_2	s_3
			s_3	s_1

Table 5 Preferences of students and priorities of course organizers

In round one of TTC, 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 that 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. Due to mutual best, s_2 has a guaranteed seat at c_1 since it is her favorite course and she has one of the q_{c_2} highest priorities at c_2 . However, under TTC, s_2 has to trade endowments with s_1 to receive that seat at c_2 , thereby allowing s_1 to obtain a seat at c_2 where she has low priority. If instead we allow s_2 to clinch her seat at c_1 without trading, we reach the outcome $\{(c_1, \{s_1, s_2\}), (c_2, \{s_3\})\}$.

4.2. Prioritized Pointing

Another property of TTC is that its pointing rule is myopic. TTC always points at the highest ranked student and allows that student to trade her priorities. If the student has very low priorities for other courses, allowing her this trade may lead to a lot of justified envy later on. However, as we will prove later, we can point to any of the guaranteed students at a course while preserving 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 that 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 for that course 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 for all of the other courses.

We use a heuristic estimate, which turns out to be very helpful in reducing the total number of instances of justified envy. First, we restrict a course c with capacity q to point to one of the q highest-priority students. This is to ensure that a trivial cycle cannot cause justified envy. Second, among this set of students, we do not use a student's priority rank at c , as this information is irrelevant. Instead, we calculate each student's average priority rank for the courses other than c .⁷ In this context, we use the term *highest average priority rank* for notational convenience, but in fact we use a more sophisticated metric. We weight priority rankings

⁷ Note that a student's priority ranking counts the number of students with which she is capable of causing justified envy. Since we have a simple objective (reduce the total instances of justified envy), this is an effective heuristic. When there is a more complicated design objective, such as weighting instances of justified envy differently, a more sophisticated heuristic would be needed.

for courses such that courses with a low remaining capacity are given high precedence while those with a high remaining capacity are given low precedence. The relative capacity of the remaining courses is a proxy for the likelihood that a course will be overdemand. Since processing a cycle changes the relative capacities of the remaining courses, these weights are dynamically adjusted.

Consider the following example:

EXAMPLE 5. Let $S = \{s_1, s_2, s_3, s_4\}$ and $C = \{c_1, c_2, c_3\}$. $q_{c_1} = 2$ and $q_{c_2} = q_{c_3} = 1$. Define preferences and priorities as follows:

\succ_{s_1}	\succ_{s_2}	\succ_{s_3}	\succ_{s_4}	\succ_{c_1}	\succ_{c_2}	\succ_{c_3}
c_1	c_3	c_1	c_3	s_4	s_2	s_1
c_2	c_2	c_2	c_1	s_2	s_1	s_2
c_3	c_1	c_3	c_2	s_1	s_3	s_3
				s_3	s_4	s_4

Table 6 Preferences of students and priorities of course organizers

In round 1 of TTC, we resolve the cycle c_1, s_4, c_3, s_1, c_1 . In round 2, s_2 is matched to c_2 and in round 3, s_3 is matched to c_1 . The final matching is $\mu = \{(c_1, \{s_1, s_3\}), (c_2, \{s_2\}), (c_3, \{s_4\})\}$.

Since s_4 is allowed to trade in the first round, she obtains a seat at c_3 , which leads to justified envy by s_2 . Note that this instance of justified envy cannot be removed by clinching, since none of the students could clinch in the first round. However, if instead of pointing at s_4 , course c_1 points at s_2 , who is also guaranteed a seat at c_1 , we reach a matching that is both Pareto efficient and fair. In this case, the first round leads to the cycle c_1, s_2, c_3, s_1, c_1 and ultimately to the matching $\mu = \{(c_1, \{s_1, s_4\}), (c_2, \{s_3\}), (c_3, \{s_2\})\}$.

Let us now introduce the Prioritized Clinch & Trade (PCT) mechanism, where we suggest a more sophisticated pointing procedure for the courses while also incorporating a clinching procedure as discussed in the previous section:

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) has a priority that 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 that 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.

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. Importantly, this mechanism is efficient,

strategyproof, and always satisfies mutual best. The proofs of the following results can be found in the appendix.

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

We discussed that no mechanism can be strategyproof, fair, and always select an efficient assignment when a fair and efficient assignment exists. However, PCT is strategyproof, fair, and efficient when there are only two courses and there is sufficient capacity for all students. A fair and efficient assignment must correspond to the student-optimal envy-free assignment. This implies that when there are two courses and sufficient capacity, PCT corresponds exactly to the student-proposing Deferred Acceptance algorithm. Interestingly, when there are N students, N courses, and each course has a capacity of one, then PCT is equivalent to TTC. However, when there are N students, two courses, and the total capacity of both courses is at least N , PCT is equivalent to the Deferred Acceptance algorithm.

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

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

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

Unfortunately, for larger assignment problems it is possible to construct instances for which traditional TTC leads to less justified envy than any of the proposed methods for increasing its fairness, such as PCT or ETTC (Hakimov and Kesten 2018). Hence, none of these algorithms lead to outcomes that can be analytically proven to be always fairer than the result of TTC. However, the positive effects of these new methods on fairness can be shown empirically. In this paper, we are mainly concerned with assignments under the additional constraint of minimum quotas. Yet, we also compared PCT and ETTC without quotas. These preliminary tests are based on field data and empirically generated data and showed that PCT resulted in less justified envy than TTC and ETTC. In our results in Section 5, we focus on the comparison different algorithms taking high and low quotas into account.

4.3. Considering Minimum Quotas

The biggest challenge we faced in creating the course assignment mechanism at our university was incorporating minimum quotas. Both clinching and non-myopic pointing are based on identifying which students are guaranteed a course. This is trivial when there are no minimum quotas; a student is guaranteed a course 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 12, 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 course 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 course 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 by 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 for all courses. Specifically, $i \succ_{ML} j$ if i has a higher average priority for 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) has a priority that 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 that 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 for courses 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 for all courses). Execute one round of TTC. If the number of assigned extended seats reaches ϵ , remove all extended courses. Update G_c according to the remaining students.

EXAMPLE 6. We use the same instance as Example 3 with the corresponding preferences and priorities shown in Table 3 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 $c_1, s_3, c_3^*, s_1, c_1$ and c_3^* is taken off the market.

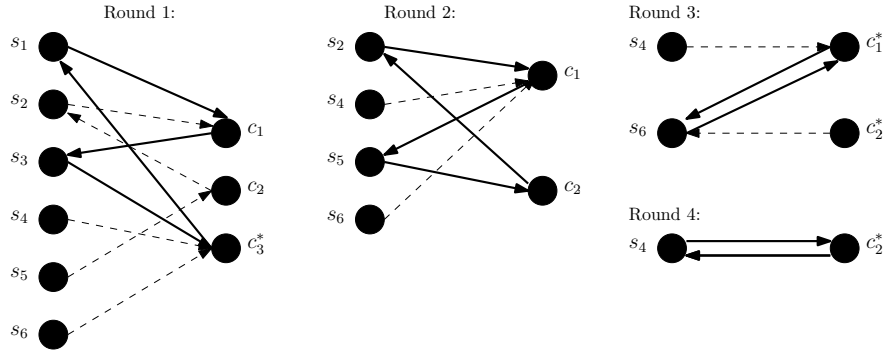


Figure 3 Run of ESPCT for Example 6

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 4.3. 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 $\mu = \{(c_1, \{s_1, s_2, s_6\}), (c_2, \{s_4, s_5\}), (c_3, \{s_3\})\}$. Since s_6 and s_4 have justified envy toward s_3 , μ contains two 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, Pareto efficient, and satisfies p -mutual best.*

Proof: See the appendix.

COROLLARY 2. *ESPCT is strategyproof, 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). In general, however, 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 $p'_c \leq p_c$ 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' , fewer students are guaranteed any individual course compared to 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 the 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 use this information for clinching and prioritized pointing. We refer to this mechanism as **Range-widening ESPCT (RESPCT)**.

4.4. 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 where 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 let G_c with $|G_c| = \sigma_c$ be the set of these students. Obviously, $\sigma_c \leq q_c$ has to hold for all courses. Furthermore, 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 the minimum quotas of other courses, i.e.

$$n - \left| \bigcup_{c \in C_0} G_c \right| \geq \sum_{c \in C \setminus C_0} p_c \text{ for all } C_0 \subset C \quad (\text{FC})$$

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). Again, all proofs for the results can be found in the appendix. 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 $p_c \leq \sigma_c \leq q_c$. Then, (FC) is satisfied.*

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

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 $\sigma = q$.*

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 in general possible 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 (MIP). 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 state whether student $s \in S$ clinches her seat at $c \in C$. Then, introduce an auxiliary variable $\lambda_c \in \mathbb{R}_{\geq 0}$ 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 of 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 .

$$\sum_{c \in C} x_{sc} \leq 1 \quad s \in S \quad (1)$$

$$x_{sc} \leq \sigma_{cs} \quad s \in S, c \in C \quad (2)$$

$$\lambda_c \leq q_c(1 - z_c) \quad c \in C \quad (3)$$

$$\lambda_c \leq p_c - \sum_{s \in S} x_{sc} + q_c z_c \quad c \in C \quad (4)$$

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

$$x_{sc} \in \{0, 1\} \quad c \in C, s \in S \quad (6)$$

$$\lambda_c \geq 0 \quad c \in C \quad (7)$$

$$z_c \in \{0, 1\} \quad c \in C \quad (8)$$

Here, constraints (1) and (2) restrict each student to clinch at most one of their guaranteed seats. All students, who 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 $\max\{0, p_c - \sum_{s \in S} x_{sc}\}$.

Unfortunately, it is not easy to make use of the MIP in order to maximize the number of guaranteed seats. One could consider 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 possibility to find vectors σ with the help of the MIP described above is by employing Algorithm 5 with $\underline{\sigma} = p$ and $S^{\text{fixed}} = \emptyset$. 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 increased. This way, after at most $\sum_{c \in C} q_c$ iterations, the algorithm returns a maximal vector σ , i.e. the number of guaranteed seats cannot be increased for any of the courses. Note that the algorithm does not necessarily find a maximum vector, i.e. one where the sum of guaranteed seats is maximized.

Algorithm 5 Greedy algorithm to determine a maximal σ vector

Input: A lower bound $\underline{\sigma}$ for σ and a set of students S^{fixed} that are already assigned to courses.

Set $\sigma = \underline{\sigma}$. Initiate a set Γ of courses c for which σ_c cannot be increased.

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' with all students from S^{fixed} who are already assigned to their courses. 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 .

4.5. 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 we instead 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 for c . Then, in the cycle resolution phase, we allow extended courses to point to 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 that has a guaranteed seat at c with the highest average priority for 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 that are pointed at by the extended courses reaches ϵ , all remaining extended courses point at the student among those ϵ with highest priority.

EXAMPLE 7. Again, we consider the same market from Example 3. Since every matching that satisfies the maximum quotas is feasible, we get $\sigma = q$. Again, $\epsilon = 3$.

Round 1: Student s_1 clinches c_1 . After that, s_6 clinches c_3^* . Student s_3 clinches her favorite remaining school c_2 . Student 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 7.

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. There is no instance of justified envy in μ .

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

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.

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

Determine σ^1 by executing Algorithm 5 with input p and define $>_{ML}$ using a student's average priority for all courses.

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) has a priority that is within σ_c^k of her favorite course c , assign s to c .

After each round of clinching, recalculate σ^k by executing Algorithm 5 with σ^k and the hitherto assigned students as input. Iteratively clinch until no student-course pair can be clinched. Define as ϵ^1 the difference between unassigned students and open seats at regular courses.

Round $k \geq 1$ - Cycle Resolution Phase: All standard and extended courses that were pointing to a student by the end of round $k - 1$ continue to point at that student. Each standard course c that was not pointing to a student in round $k - 1$ points to the student in G_c^k with the highest average priority for courses other than c . For each extended course, c , if $\sigma_c^k > p_c^k$ and c was not pointing to a student at the end of round $k - 1$, then c points to the student in G_c^k with the highest average priority for courses other than c . If all extended courses point to fewer than ϵ^k many students, iteratively select additional extended courses and have them point to their highest prioritized student. Specifically, the extended course that represents course c points to the student in G_c^k with the highest average priority at courses other than c . If extended courses already point to ϵ^k students, let all other extended courses point to the student among those ϵ^k who has the highest priority according to $>_{ML}$ (i.e., the highest average priority for all courses). Execute one round of TTC. Calculate σ^{k+1} via Algorithm 5 for an input of σ^c and the hitherto assigned students. Define as ϵ^{k+1} the difference between unassigned students and open seats at regular courses. When $\epsilon^{k+1} = 0$, remove all extended courses.

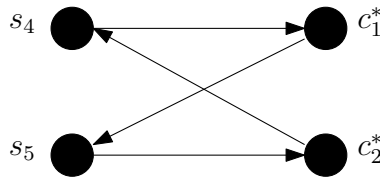


Figure 4 Run of RESPCT for Example 7

5. Evaluation

In order to evaluate the level of envy caused by the RESPCT mechanism discussed in this paper, we conducted a computational study using field data from the computer science department at the Technical University of Munich.

5.1. Data and Experimental Design

In the following, we briefly describe the field data from the course assignment application at the Technical University of Munich. Students and lecturers participate in this mechanism – students by stating their preferences over courses and lecturers by stating the capacity of their course and their preferences over students. The registration data sets describe preferences for seminars and practical courses for both bachelor’s and master’s students collected between June 2014 and March 2016. For our experiments, we only considered data sets where the number of students does not exceed the total course capacity but does exceed the sums of minimum quotas we introduced. These sets range from 27 to 733 different students and six to 43 courses (see Table 7).

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

Table 7 Summary of two-sided data sets

Let us provide a few additional statistics on the distribution of preferences and highlight 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. For each course in the two data sets Figure 5 shows the fraction of students that rank the corresponding course as their top choice, or among their top three (or five) choices, respectively. The chart shows the heterogeneity of the students’ preference lists except for a few popular courses, especially when considering the top five choices. Even in TS2 with only six courses, no course was ranked within the top five 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 three priorities and 106 students are ranked among the top five, with a maximum number of $26 \cdot 3 = 78$ and $26 \cdot 5 = 130$ spots available. For TS2, five distinct students are ranked as top priority, 14 within the top three and 18 within the top five, with a maximum of six, 18, and 30 spots available, respectively.

Course organizers submitted only maximum capacities, since minimum quotas were not yet part of the course assignment mechanisms. Therefore, in order to evaluate the mechanisms for each of the data sets we tested a range of different minimum quotas. Based on our experience with the matching system used at the

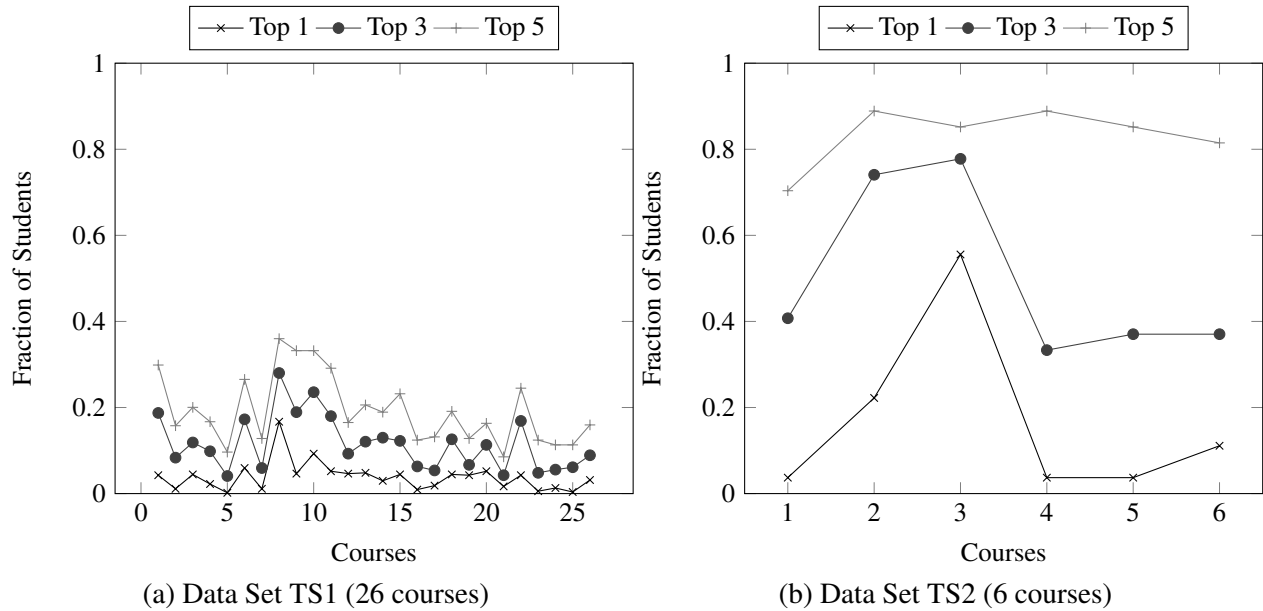


Figure 5 Fraction of students that have a course in the top 1,3 or 5 of their preference lists

university, we consider a range from 3 to 7 as realistic 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 , 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 most fair 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.

5.2. Fairness

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

We will now discuss the degree of (un-)fairness in 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 the *instances of justified envy per student* for each minimum quota vector, averaged over all 10 instances. RESPCT performed best among all mechanisms, even with a lower number of guaranteed

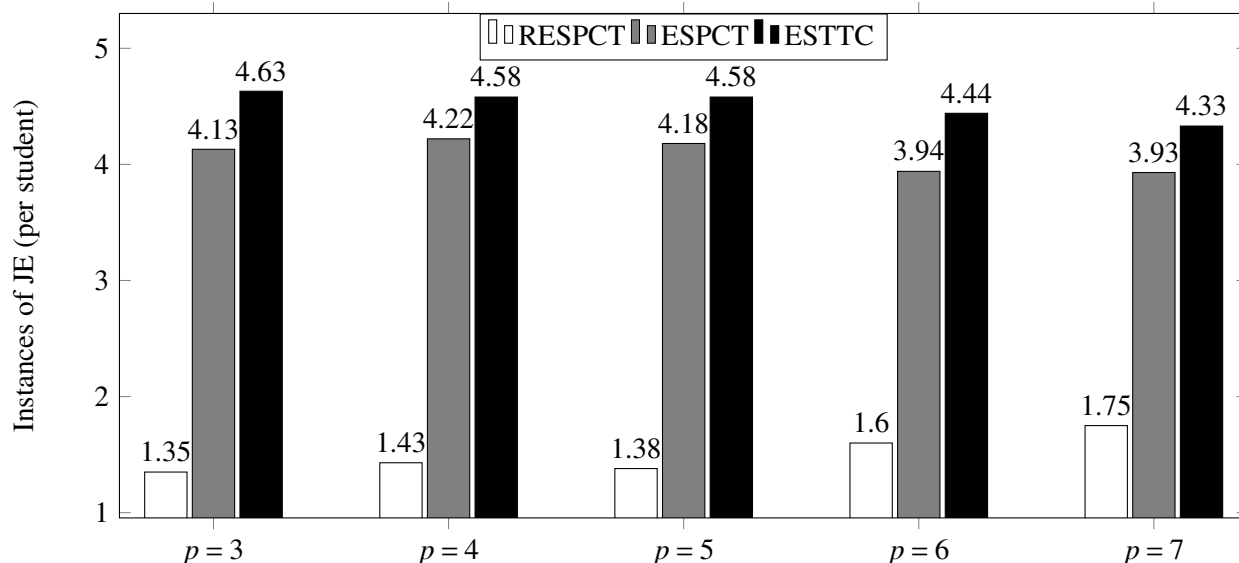


Figure 6 Instances of justified envy (per student) when applying the respective matching mechanisms

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 mean that fewer students can be guaranteed a course. This reduces the impact of both clinching and prioritized pointing. Paradoxically, the performances of ESTTC 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 the master list and not course priorities, 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.

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

5.3. 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. (2012). ESDA led to very inefficient outcomes: On average, roughly 10 percent of students could be Pareto-improved under the envy-free matching obtained by ESDA. Figure 9 depicts the average rank of all students, and Table 4 shows the distribution of ranks for RESPCT and ESDA. It can be seen that RESPCT

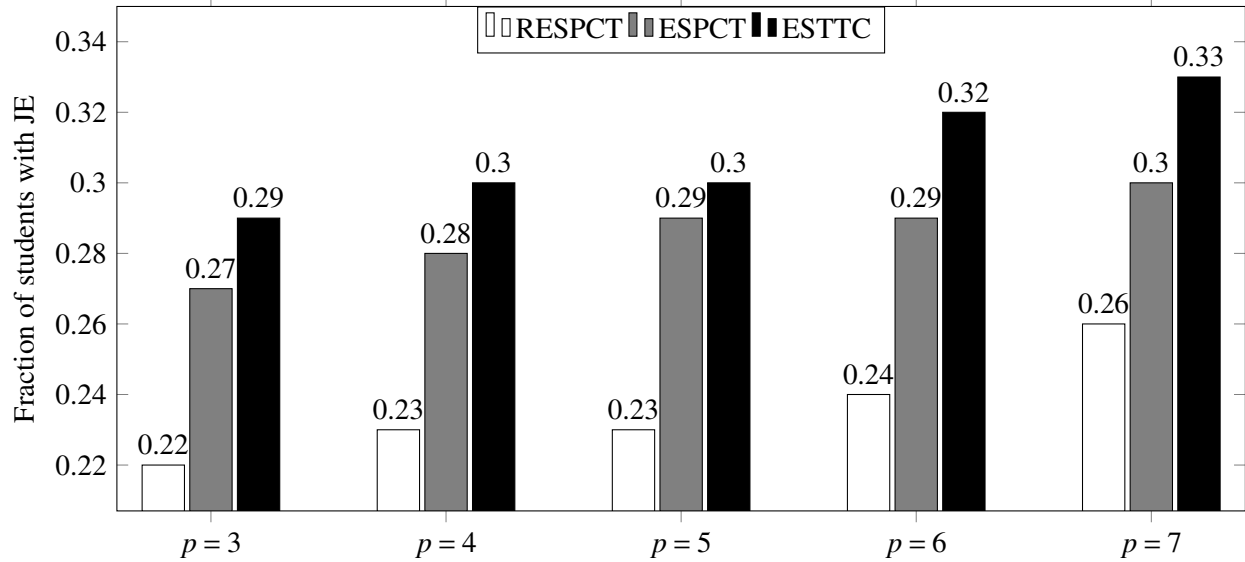


Figure 7 Fraction of students that have justified envy when applying the respective matching mechanisms

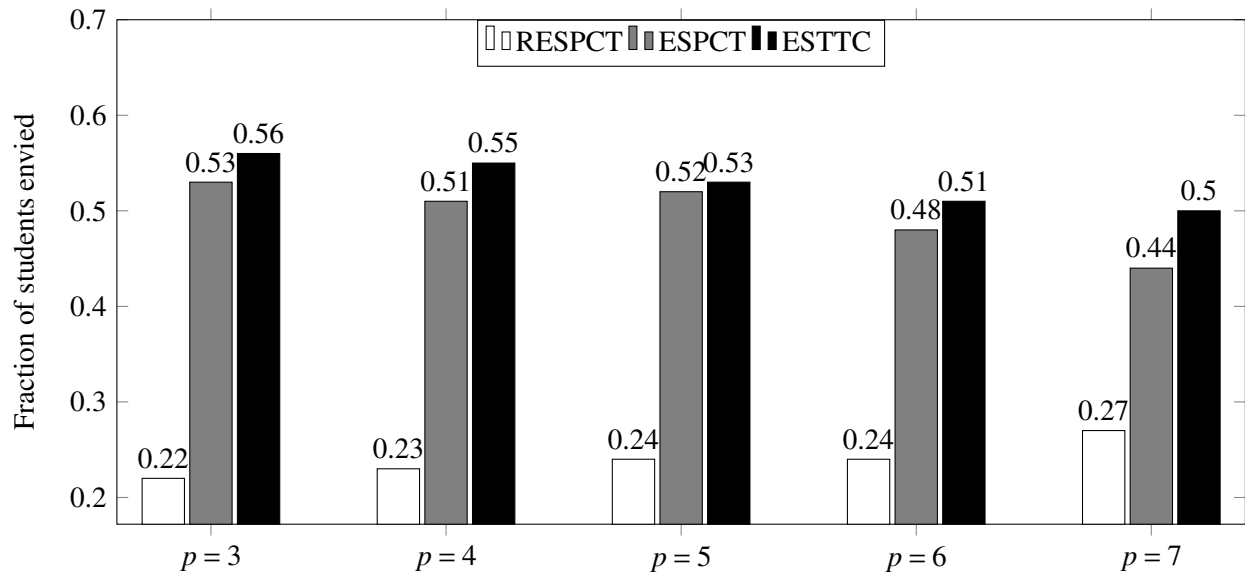


Figure 8 Fraction of students that are envied under the respective matching mechanisms

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 8, we report the rank distributions and fairness measures for all mechanisms discussed in this work for the illustrative case of $p = 5$. It can be observed once again that the TTC variants are much more efficient than ESDA with RESPCT being the fairest of the efficient mechanisms.

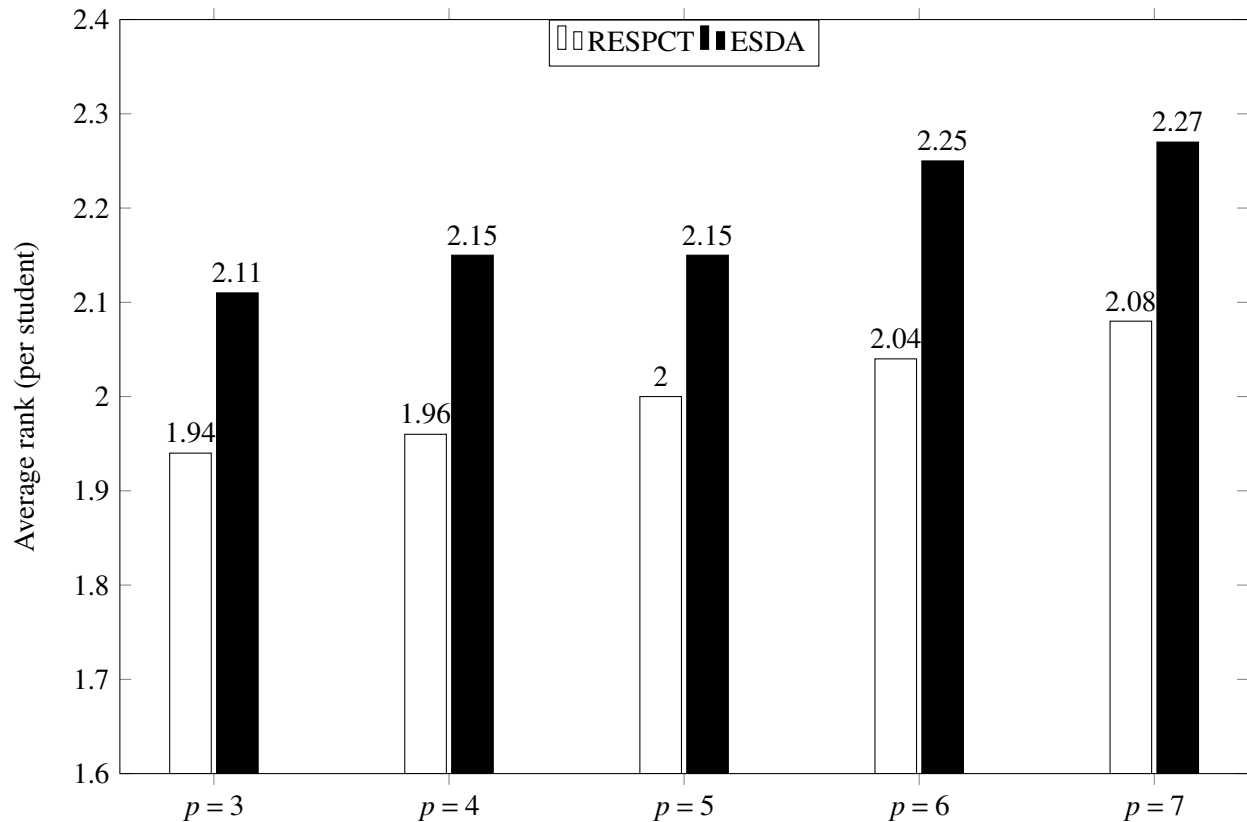


Figure 9 Average rank (per student)

Rank	p = 3		p = 4		p = 5		p = 6		p = 7	
	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 8 Comparison of efficiency between RESPCT and ESDA

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

5.4. Scalability

In order to evaluate how well our algorithms scale, we generated synthetic data sets in addition to our field data. We created matching instances with up to 2,000 students and 60 to 100 courses of varying

minimum quotas and capacities. The preferences of students and courses were designed in such a way as to resemble the structure of real-world preference profiles we witnessed in the field data. While the runtime of RESPECT increased significantly, we were still able to solve even the largest instances within 90 minutes. Since the course assignment only has to take place once at the beginning of each semester, these runtimes are still acceptable even for this large number of participants. The results for these larger instances confirm the results we obtained for the field data: RESPECT clearly outperforms all other efficient algorithms with regards to fairness and is significantly more efficient than ESDA.

6. Discussion

Coordination has long been a central topic in the decision support and design science literature, where economic design desiderata matter (Banker and Kauffman 2004). Auction mechanisms are an integral part of this literature, but monetary transfers are often not an option. Matching with preferences is another essential pillar of the broader market design literature, and it provides an important mechanism for the design of information systems. We draw on recent results of matching with preferences and make a design science contribution to address one of the central problems in this literature: the trade-off between efficiency and fairness. While TTC (or its quota-respecting version ESTTC) incurs justified envy, DA (or ESDA) leads to very inefficient assignments. The level of inefficiency is substantial, which leads to a challenging trade-off for designers.

We introduced the RESPCT mechanism. By incorporating a non-myopic pointing rule for the courses and maximizing the number of guaranteed seats at each course, RESPCT assigned students in a significantly fairer way than ESTTC. It is the combination of these innovations that makes a substantial difference in the level of justified envy. RESPCT outperforms ESTTC by far regarding the fairness metrics that we studied. Experiments show that even without widening the range of guaranteed seats and when only employing clinching and non-myopic pointing, the resulting ESPCT algorithm still has significant advantage over ESTTC, which confirms our expectations regarding the fairness of the two mechanisms.

If fairness is the only concern of the designer, DA-based algorithms like ESDA are a clear choice. However, efficiency almost always matters and designers do not want to be wasteful. RESPCT is fully efficient and has very little envy. Especially when the ratio of agents required to fill quotas within the number of overall agents is low, RESPCT outperforms the other algorithms since all of its building blocks can be made use of. Most importantly, although we have assumed objects have minimum quotas, the advantages of RESPCT over TTC (when there are no minimum quotas) are even greater than the advantages of RESPCT over ESTTC (when there are minimum quotas). Without minimum quotas, it is trivial to calculate which agents are guaranteed an object,⁸ so RESPCT retains all of the advantages of clinching and non-myopic pointing but without the computational burden. When there are minimum quotas, RESPCT is best suited

⁸ Without a minimum quota, an agent is guaranteed an object with capacity q if she has one of the q highest priorities.

for applications, such as course assignment, where the minimum requirement is small relative to the overall capacity of the object.

RESPCT can easily be implemented for any matching problem that can be solved via (ES)TTC or DA mechanisms. In order to calculate the vector of guarantees, RESPCT should have access to a mathematical programming solver. In our experiments we have worked on a large-scale matching problem with 2,000 students and 100 courses. These instances could all be solved within 90 minutes on commodity hardware. Larger assignment problems in domains such as course assignments, cadets-to-branch matching, and hospital-resident matchings are usually performed only once every few months and such computation times should not constitute a problem. Hence, RESPECT can serve as a remedy for many one-to-many assignment problems.

As always, there are limitations. RESPCT is not well suited for environments with complicated admissions requirements. For example, college admissions in India involve a complicated interaction of horizontal and vertical reserve requirements (Sönmez and Yenmez 2019). In such an environment, determining which students are guaranteed admission to a university would likely be computationally infeasible. Other limitations stem from the type of preferences that can be elicited from participants. RESPCT is designed for ordinal preferences and priorities that are complete, irreflexive, and transitive. Moreover, preferences are independent and private and do not change when students observe the preferences of others. In cases where the students' preferences and their satisfaction with an outcome depend on these or other factors, a designer might be better advised to model the assignment as a mathematical optimization problem and search for an efficient solution. However, by doing so, the designer loses strategyproofness and it might not even be possible to find an envy-free allocation if the preferences of individuals change once the number of interested students is revealed. For example, Jehiel et al. (2006) showed that, in environments with interdependencies, only constant rules, according to which the same alternative is chosen for every type realization, are incentive compatible.

A key assumption in most of the literature on one-to-many assignment problems is that of unit demand, i.e. students want to win only one out of many course seats. It turns out that multi-object assignment problems, e.g. the assignment of course schedules to students, is a much harder problem. The only strategyproof mechanisms available are serial dictatorships Ehlers and Klaus (2003), Pápai (2001). Only recently have randomized mechanisms been introduced that satisfy weaker notions of incentive-compatibility (Nguyen et al. 2016).

7. Conclusions

The trade-off between fairness and efficiency leads to a significant policy problem in many applications of matching with preferences. Fairness is often prioritized over efficiency (Roth 2002). We show that with appropriate algorithms one can implement fully efficient results with very low levels of envy. This might well be an argument in favor of TTC in other applications such as school choice, where the topic has been

intensively discussed over the years (Abdulkadiroğlu and Sönmez 2003). Also in school choice minimum quotas matter, and regulators want to make sure that all schools seats are filled even if they are less popular.

Overall, we provide a strategyproof and efficient mechanism with very low levels of envy, which was the main argument for the computer science department at the Technical University of Munich to switch from the DA algorithm to RESPCT. We argue that such considerations will also apply to other domains, such as medical labor markets, cadet matching in the military, or school choice, where there has been a long debate about DA vs. TTC. RESPCT is a versatile tool for one-to-many object assignments without monetary transfers and a powerful means for the design of distributed information systems. The unit demand assumption in assignment problems is the most severe but also the most challenging limitation, one that is currently drawing significant attention (Budish and Cantillon 2012, Nguyen et al. 2016, Karaenke et al. 2020), as it would considerably extend the number of applications for matching with preferences.

Acknowledgments

We thank the anonymous reviewers and participants of the Auction and Market Design cluster at the INFORMS Annual Meeting for the many valuable comments. We are grateful for funding by the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft, BI 1057/1-8).

References

- 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.
- Adomavicius, G., S. Curley, A. Gupta, P. Sanyal. 2013. Impact of information feedback in continuous combinatorial auctions: An experimental study of economic performance. *MIS Quarterly* **37** 55–76.
- Adomavicius, G., A. Gupta. 2005. Toward comprehensive real-time bidder support in iterative combinatorial auctions. *Information Systems Research (ISR)* **16** 169–185.
- Ba, Sulin, Jan Stallaert, Andrew B Whinston. 2001. Research commentary: introducing a third dimension in information systems design—the case for incentive alignment. *Information systems research* **12**(3) 225–239.
- Balinski, Michel, Tayfun Sönmez. 1999. A Tale of Two Mechanisms: Student Placement. *Journal of Economic Theory* **84**(1) 73–94. doi:<http://dx.doi.org/10.1006/jeth.1998.2469>.
- Banker, Rajiv D, Robert J Kauffman. 2004. 50th anniversary article: The evolution of research on information systems: A fiftieth-year survey of the literature in management science. *Management Science* **50**(3) 281–298.
- Bapna, R., S. A. Chang, P. Goes, A. Gupta. 2009. Overlapping online auctions: Empirical characterization of bidder strategies and auction prices. *MIS Quarterly* **33**(4) 763–783.
- Bapna, R., P. Goes, A. Gupta. 2003. Replicating online yankee auctions to analyze auctioneers' and bidders' strategies. *Information Systems Research* **14**(3) 244–268.

- Bichler, Martin, Alok Gupta, Wolfgang Ketter. 2010. Research commentary—designing smart markets. *Information Systems Research* **21**(4) 688–699.
- Bichler, Martin, Zhen Hao, Gediminas Adomavicius. 2017. Coalition-based pricing in ascending combinatorial auctions. *Information Systems Research* **28**(1) 159–179.
- Biró, Péter. 2008. Student admissions in Hungary as Gale and Shapley envisaged. *University of Glasgow Technical Report TR-2008-291* .
- Biró, Péter. 2017. Applications of matching models under preferences .
- Budish, Eric, Gérard P Cachon, Judd B Kessler, Abraham Othman. 2016. Course match: A large-scale implementation of approximate competitive equilibrium from equal incomes for combinatorial allocation. *Operations Research* **65**(2) 314–336.
- Budish, Eric, Estelle Cantillon. 2012. The multi-unit assignment problem: Theory and evidence from course allocation at harvard. *American Economic Review* **102**(5) 2237–71.
- Delacrétaz, David, Scott Duke Kominers, Alexander Teytelboym. 2016. Refugee resettlement. *University of Oxford Department of Economics Working Paper* .
- Diebold, Franz, Haris Aziz, Martin Bichler, Florian Matthes, Alexander Schneider. 2014. Course Allocation via Stable Matching. *Business & Information Systems Engineering* **6**(2) 97–110. doi:10.1007/s12599-014-0316-6. URL <http://dx.doi.org/10.1007/s12599-014-0316-6>.
- Diebold, Franz, Martin Bichler. 2017. Matching with indifference: 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.
- Ehlers, Lars, Bettina Klaus. 2003. Coalitional strategy-proof and resource-monotonic solutions for multiple assignment problems. *Social Choice and Welfare* **21**(2) 265–280.
- Fan, Ming, Jan Stallaert, Andrew B Whinston. 2003. Decentralized mechanism design for supply chain organizations using an auction market. *Information Systems Research* **14**(1) 1–22.
- 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, Atsushi 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.
- Gregor, Shirley, Alan R Hevner. 2013. Positioning and presenting design science research for maximum impact. *MIS quarterly* 337–355.

- Hakimov, Rustamdjan, Onur Kesten. 2018. The equitable top trading cycles mechanism for school choice. *International Economic Review* .
- Hevner, Alan, Samir Chatterjee. 2010. Design science research in information systems. *Design research in information systems*. Springer, 9–22.
- Hevner, Alan, Salvatore T March, Jinsoo Park, Sudha Ram. 2004. Design science in information systems research. *MIS quarterly* **28**(1) 75–105.
- Hevner, Alan R. 2007. A three cycle view of design science research. *Scandinavian journal of information systems* **19**(2) 4.
- Jehiel, Philippe, Moritz Meyer-ter Vehn, Benny Moldovanu, William R Zame. 2006. The limits of ex post implementation. *Econometrica* **74**(3) 585–610.
- Kamada, Yuichiro, Fuhito Kojima. 2015. Efficient matching under distributional constraints: Theory and applications. *American Economic Review* **105**(1) 67–99.
- Karaenke, Paul, Martin Bichler, Soeren Merting, Stefan Minner. 2020. Non-monetary coordination mechanisms for time slot allocation in warehouse delivery. *European Journal of Operational Research* .
- Liu, De, Jianqing Chen, Andrew B Whinston. 2010. Ex ante information and the design of keyword auctions. *Information Systems Research* **21**(1) 133–153.
- Lu, Yixin, Alok Gupta, Wolfgang Ketter, Eric van Heck. 2016. Exploring bidder heterogeneity in multichannel sequential b2b auctions. *MIS Quarterly* **40**(3) 645–662.
- Lu, Yixin, Alok Gupta, Wolfgang Ketter, Eric van Heck. 2017. Information transparency in b2b auction markets: The role of winner identity disclosure. *Available at SSRN 2949785* .
- Ma, Jinpeng. 1994. Strategy-proofness and the strict core in a market with indivisibilities. *International Journal of Game Theory* **23**(1) 75–83.
- 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.
- Nguyen, Thanh, Ahmad Peivandi, Rakesh Vohra. 2016. Assignment problems with complementarities. *Journal of Economic Theory* **165** 209–241.
- NRMP. 2014. National Resident Matching Program. <http://www.nrmp.org>. [Online; accessed 19-August-2014].
- Pápai, Szilvia. 2000. Strategyproof assignment by hierarchical exchange. *Econometrica* **68**(6) 1403–1433.
- Pápai, Szilvia. 2001. Strategyproof and nonbossy multiple assignments. *Journal of Public Economic Theory* **3**(3) 257–271.
- Pycia, Marek, M Utku Ünver. 2017. Incentive compatible allocation and exchange of discrete resources. *Theoretical Economics* **12**(1) 287–329.

Roth, Alvin E. 1982. 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.

Shapley, Lloyd, Herbert Scarf. 1974. 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.

Sönmez, Tayfun, Bumin Yenmez. 2019. Affirmative action in india via vertical and horizontal reservations. *Working paper* .

Sun, Ning, Zaifu Yang. 2003. A general strategy proof fair allocation mechanism. *Economics Letters* **81**(1) 73–79.

Proofs

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

Proof: Pareto efficiency: Each student in round 1 is assigned to her most preferred course and therefore cannot be made strictly better off. If a student i in round 2 is not assigned her most preferred course, then that course was assigned to capacity in round 1. Any reassignment that makes i strictly better off must make a student from round 1 strictly worse off. Iterating this argument, we find that no student assigned in round k may be made strictly better off without making a student assigned in an earlier round strictly worse off. Therefore, the algorithm is *Pareto efficient*.

Strategyproofness: We first discuss the trading phase of the mechanism: Consider any student i and preference profile \mathcal{P} of the students. We fix the preferences of the students other than i , and at the start of each round, we allow i to point at whatever course she wishes. Suppose for contradiction that there is some round where i does strictly better by pointing at a course other than her most preferred course, and let k be the last round i is able to benefit by misreporting. If i does not form a cycle in round k , then her report in this round does not effect the outcome, and she may make the same misreport in the next round and receive the same assignment. This contradicts k being the latest round she is able to profitably misreport her preferences. If i does form a cycle by pointing at a course b even though her most preferred course a still has available capacity, then there is a path from b to i . If i had pointed at her true favorite a instead of b , then she could not have formed a cycle, as otherwise pointing at b is not be a profitable misrepresentation. Therefore, if i points to a in round k , b is not assigned in round k and indeed the entire path from b to i is preserved through round $k + 1$. Therefore, i can make the same profitable misrepresentation in round $k + 1$ as in round k , which contradicts k being the last round in which i is able to profitably misreport her preferences. Therefore, i never does better than pointing at her most preferred course.

Clinching does not interfere with strategyproofness: 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 a to a student i , then that path remains until i is assigned. Therefore, i 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 a if she is guaranteed admissions to a , and once a student is guaranteed admissions to a course, then she never loses that status.

Mutual best: The analysis above also implies that PCT satisfies mutual best as a student is never assigned a course worse than one she is guaranteed admission to. Q.E.D.

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

Proof: Label the two courses 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 $\mu(j) = b$, $bP_i a$, and $i >_b j$. Student j could not have clinched b given that, since i has higher priority and b is i 's most preferred course, 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 course with available capacity and b must have had available capacity when j was assigned, i could not have been assigned before j . Note that as there are only two courses, the

highest average priority at the courses other than a corresponds exactly 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, Pareto efficient, and satisfies p -mutual best.

Proof: The proofs for the first two properties are due to Fragiadakis et al. (2012). We only show p -mutual best: Denote by 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 the standard course c in the extended market, s holds one of the p_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, the student in P_c who holds the top priority is assigned a course other than c . In the next round, c points to the next student in P_c . Since $|P_c| \leq p_c$, course 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,

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

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,

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

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 prove strong NP -hardness by reduction from vertex cover. Let $G = (V, E)$ be an undirected graph and let k be 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, \dots, d_α .

Define the 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, \dots, 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}, \dots, c_{v_k} . Thus for $C_0 = \{c_{v_1}, \dots, c_{v_k}\}$, it holds that

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

leading to an incompatibility of σ and p .

Suppose now that σ is not compatible with p . Since α students are not guaranteed a seat at any course and all courses have a minimum quota of 1, there must exist a subset of courses C_0 with a size of 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 RESPCT. Furthermore, no student can receive a course by clinching that she could not have otherwise received: Denote by G_c^k the set of students who are guaranteed a seat at course c in step k . Assume student $s \in S$ can receive a seat at regular course $c \in C$ during the clinching phase of some step of RESPCT. Thus, $s \in G_c^k$ 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 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. This case is equivalent, as in ESTTC, and hence strategyproofness holds as in 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 by Z^{*k} the set of students who are endowed a seat at an extended course at the beginning of round k . Denote by $Z^k(s, d)$ 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^* that 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 that is different from her most preferred course 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 of Pareto efficiency is equivalent to the proof for ESTTC.

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 4.4 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.