FAIRNESS COMPARISONS OF MATCHING RULES

Pooya Ghasvareh

A thesis
in
The Department
of
Economics

Presented in Partial Fulfillment of the Requirements
For the Degree of
Doctor of Philosophy (Economics)
Concordia University
Montréal, Québec, Canada

Spetember 2019

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: **Pooya Ghasvareh**

Entitled: **Fairness Comparisons of Matching Rules**

and submitted in partial fulfillment of the requirements for the degree of

Doctor Of Philosophy (Economics)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

______________________________ Chair
Dr. Amy Swiffen

______________________________ External Examiner
Dr. Vikram Manjunath

______________________________ External to Program
Dr. Jamal Bentahar

______________________________ Examiner
Dr. Effrosyni Diamantoudi

______________________________ Examiner
Dr. Huan Xie

______________________________ Thesis Supervisor
Dr. Szilvia Papai

Approved by ______________________________
Dr. Christian Sigouin, Graduate Program Director

October 31, 2019 ______________________________
Dr. Andre Roy, Dean
Faculty of Arts and Science

Abstract

Fairness Comparisons of Matching Rules

Pooya Ghasvareh, Ph.D.

Concordia University, 2019

This thesis consists of three studies in matching theory and market design. Its main focus is to compare matching rules according to normative criteria, primarily fairness, when objects have priorities over agents.

In the first study we analyze one-to-one matching and prove that in general we cannot find a strategy-proof and Pareto-efficient mechanism which stands out uniquely in terms of fairness when using fundamental criteria for profile-by-profile comparison. In particular, despite suggestions to the contrary in the literature, the Top Trading Cycles (TTC) mechanism is not more fair than all other mechanisms in this class. We also show that while the TTC is not dominated, if the priority profile is strongly cyclic then there is not much scope for TTC to dominate other matching rules in this class.

In the second study, which focuses on many-to-on matching, I provide a direct proof that Ergin's cycle (Ergin, 2002) is stronger than Kesten's cycle (Kesten, 2006), due to different scarcity conditions for the quotas on objects. I also prove that when there is a Kesten cycle there is no strategy-proof and Pareto-efficient mechanism which uniquely stands out in terms

of the fairness criteria. Moreover, I use simulations to show that as the number of Kesten cycles increases, there are more fairness violations and fewer preference profiles at which the TTC mechanism is fair.

The third study compares three competing many-to-one matching mechanisms that are strategy-proof and Pareto-efficient but not fair, namely the TTC, Equitable Top Trading Cycles (ETTC) and Clinch and Trade (CT) mechanisms. Although one would expect that ETTC and CT are more fair than the TTC, I demonstrate the opposite for specific preference profiles and compare the aggregate number of fairness violations using simulations. I find that ETTC tends to have fewer priority violations in the aggregate than the other two mechanisms across both different quota distributions and varying correlations of preferences. Finally, I show that all three mechanisms become more efficient when the commonly most preferred object has the highest quota, and demonstrate that the more unequal the quota distribution, the more fair and efficient the three mechanisms become.

Acknowledgments

I would like to express my profound gratitude to my supervisor, Dr. Szilvia Pápai. She goes above and beyond, and her insightful comments have helped me through the process of the thesis writing and finding a job. She was always available and helped me to improve and refine my thesis.

Besides my advisor, I would like to thank the rest of my thesis committee for spending their precious time on reviewing this work and attending the defence: Dr. Vikram Manjunath, Dr. Jamal Bentahar, Dr. Effrosyni Diamantoudi, Dr. Huan Xie and Dr. Amy Swiffen.

I would also like to thank my first Graduate Program Director, Dr. Effrosyni Diamantoudi, who has supported me on several occasions, especially in the job market, and Dr. Huan Xie, for reading one of the chapters a few months before the defence and giving me constructive comments.

There are many friends who were very supportive during my cancer treatment and during the years when I was not yet diagnosed but was suffering from shortage of breath: Ehsan Fazel, Hanif Rashtian, Alireza Omidian, Paria Agharabi, Shahab Tasharrofi, Hanieh Edalati, Ali Abolghasemi, Nasim Tavassoli, Kamran Afzali, Sanaz Souri, Amirhossein Barzinpour, Noemie Cabau, Widad Achargui, Liang Wang, Olha Hnatyshyn, Tokunbo

Aromiwura, Babak Afshar, Ehsan Jafari, Bahman Bakhtiari, Mina Shahtaheri, Golnaz Davalloo, Dilek Sayedahmed, Cyrille Nsengiyumva, Benjamin Treitz and Jean Hans Garcon. I am very thankful to them for their advice, sympathy and company which helped me keep a positive attitude during the fight against cancer. My sister, Pooneh, was very helpful during some tough times that I had faced. She left her job and came to Montreal for many days to help me.

I also want to extend my gratitude to the passionate staff who were helping me in the department and answering my countless questions. Among them: Mrs. Elise Melancon, who is still dealing with graduate students, and Mrs. Lucy Gilson and Mrs. Lise Gosselin, both retired, who were always available for us. The new secretarial team that has been settled in the department was also very helpful on many occasions. I want to thank Mrs. Melissa Faisal and Mrs. Sandra Topisirovic who were always helpful and kind.

And finally, and most importantly, I would like to thank my beloved spouse, Ghodsi Madani, for standing by me and helping me during this lengthy procedure of studying and the defence, and for all her support during chemotherapy. She has been beside me for more than twelve years and I wouldn't be here without her help and support.

Contribution of Authors

Chapter 2 of the thesis, Fairness Comparisons of Strategy-Proof and Efficient Matching Rules, is a joint work with my supervisor, Dr. Szilvia Pápai.

Chapters 3 and 4 are not co-authored.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Matching Theory

Matching theory provides a new way to study supply and demand in some markets. Thinness of the market, fairness, and truth-telling are problems in markets where participants' identity matters. All of these have been addressed by matching theory, which studies centralized markets where participants submit their preferences to the market designer who decides how to match the two sides of the market.

There are different types of matching models. There is one-sided matching (which nonetheless has two sides) in which on one side of the market are agents and on the other side are objects, where the latter are always indivisible items. For example, professors are matched to offices, or students to schools. Objects have priorities which may be imposed by law. For example, neighborhood schools may have to prioritize students who live nearby. Agents are typically human beings who have different information and different tastes, which forms their preferences over the objects. There is also two-sided matching in which on both sides of the market there are agents, such as medical students and hospitals with residence positions.

Matching theory studies requirements of efficiency, fairness and incentives, in order to lay the foundations for the more practical analyses of market design, which seek to improve the matching outcomes in real-world matching markets. Normative and incentive criteria are defined specifically for matching markets, and the formal definitions may differ from the way we think of these concepts in everyday life. Examples of unfairness in real life are numerous; for instance, cronyism or lobbying, or any sort of favoritism are considered unfair, and more generally inequality without a basis is deemed unfair. In matching the definition of fairness is very specific, and in our context where objects have priorities over agents is based on the specified formal model. Namely, it is considered unfair if an agent is assigned an object which another agent with a higher priority for this object envies, given the allocation of the higher-priority agent. This is called *justified envy*, as opposed to envy that is not justified based on the priorities, and fairness calls for eliminating all instances of justified envy for all different configurations of priorities and preferences.

In the two-sided model where both sides have agents, justified envy is identical to having two agents on the two different sides of the market mutually like each other compared to their matches, and the two agents are said to form a blocking pair. When all such blocking pairs are absent for arbitrary priorities and preferences, the matching is considered stable.[1] Thus, depending on the interpretation and applications of the model (whether one-sided or two-sided) fairness and stability are closely related concepts and are identical in terms of the formal definitions.

Pareto-efficiency considers optimizing the allocation of all agents in the market in the following sense: if the allocation made by a rule is Pareto-efficient, then the only way we can improve the allocation of one agent is at the cost of making another agent worse off. Therefore, Pareto-efficiency ensures that agents' welfare in terms of their preference rankings cannot be further increased without harming at least one agent. This concept is used widely in economics, and it may be different from efficiency concepts used elsewhere in a broader sense or in other contexts. We also use a different measure of efficiency in this thesis in addition to Pareto-efficiency, where we consider the ranking of the allocations received by agents.

In order to provide the correct incentives to agents, a matching rule is often required to satisfy *strategy-proofness,* a notion which guarantees that individuals have no incentive in the matching procedure to misrepresent their preferences to the market designer, as they cannot obtain a better allocation by doing so. This is a demanding concept, and less demanding incentive constraints may also be studied, along with concepts that are unrelated to strategy-proofness but address other aspects of incentives. One example of strategic behavior without manipulating reported preferences is when agents try to show that they are more competent than they are in

[1]Stability is also closely related to the *core*, which is the central solution concept in cooperative game theory. See more on this in Roth and Sotomayor (1990 [35]).

reality, in order to get higher rankings in priority orderings. Ruling out this kind of strategic behavior is logically unrelated to strategy-proofness. In this thesis, we will focus on strategy-proofness when considering incentives. It is important to point out that strategy-proofness *does not* aim to ensure that preferences are reported truthfully merely for its own sake. A more far-reaching implication of strategy-proofness, given that the normative properties of matching rules (such as fairness and efficiency) are applied to the reported preferences, is that without truthful reporting we essentially lose the normative properties as well (or at least lose account of them), in addition to subjecting agents to risky outcomes and forcing them to make strategic calculations.

Market design, building primarily on matching theory and using the additional tools of laboratory experiments, simulations and empirical analysis as applicable, has been developed mainly since the 1990's. Its growing success led to awarding the Noble prize in 2012 to Alvin E. Roth and Lloyd S. Shapley for the theory of stable matching and the practice of market design. Despite the brief history of market design, it has touched the lives of many people in positive ways and has been widely used for many real-world allocation problems. Applications of matching theory range from job market assignments (such as medical residents who need to be matched to hospitals) to kidney allocation and refugee matching (e.g., [33], [34], [18]), among many other potential applications.

1.2 Two Prominent Matching Mechanisms: DA and TTC

There are two important matching mechanisms (also called matching rules) that have been introduced and widely studied: Deferred Acceptance (DA) and Top Trading Cycles (TTC). We present descriptions of these mechanisms here in a one-to-one one-sided matching model where objects have strict priorities over agents and agents have strict preferences over objects. Strict priorities and preferences mean that indifferences are not allowed, and thus if there is a choice of two agents then the higher-priority agent for each object is unambiguously defined, and if there are two objects then for each agent one is preferred to the other one (but not the other way around).

Steps for the DA rule (Gale and Sapley, 1962 [14]):

- Each agent proposes to their most preferred object.
- Each object keeps only the highest priority proposing agent tentatively and rejects the rest.
- All rejected agents propose to their next preferred object and the process is repeated iteratively.
- The algorithm ends when all agents are either matched or prefer to stay unmatched.

Steps for the TTC rule (Shapley and Scarf, 1974 [37]; Pápai, 2000 [29]):

- Each object forms a pair with its highest priority agent who is in the market.
- Each agent points to the pair with their most preferred object (which may be their own pair).
- Since there is a finite number of agents, there is going to be at least one cycle formed by pointing agents which, if the corresponding trade is carried out, improves the allocation of the involved agents (or leaves the allocation unchanged if an agent points to their own pair).
- Agents are permanently matched to the objects according to the cycles.
- Matched agents and objects are removed from the market and the process is repeated iteratively.
- The algorithm ends when all agents are either matched or prefer to stay unmatched.

The DA and TTC mechanisms are well-studied because they possess notable attributes. First, both the DA and TTC are strategy-proof in the one-sided matching model. Therefore, if any of the agents misreports their ranking, there is no way for them to get a better allocation. In addition, both of these rules has a property that does not hold for the other one. The DA is fair ([14]) which means, as already explained, that if an agent gets matched to an object then there is no other agent who has higher priority for the object and prefers this object to their current allocation. However, the DA is not Pareto-efficient, although it has been shown that the DA allocation Pareto-dominates all other fair allocations ([14]). On the other hand, the TTC is Pareto-efficient ([37], [29]) for the agents, which means that agents cannot get a better allocation unless at least one agent is made worse off. However, the TTC is not fair.

1.3 Objectives and Motivation

In this thesis we study the TTC and alternative rules to the TTC which are strategy-proof and Pareto-efficient but not fair, both in one-to-one and many-to-one matching models. One of the main aims of this thesis is to find matching rules that perform relatively better than others by having fewer justified envy instances of assignments when taking into account the priorities of the objects.

Gale and Shapley (1962 [14]) and Balinski and Sönmez (1999 [7]) demonstrate that it is not possible to have an outcome that is both Pareto-efficient and fair at each priority and preference profile. To see this, consider the preferences and priorities provided in Table 1(a)-(b). It is easy to verify that the unique fair allocation is the one given in Table 1(c). However, this allocation is not Pareto-efficient, since it is Pareto-dominated by the allocation displayed in Table 1(d).

This is a classic example to show the incompatibility of fairness and efficiency in a simple matching market, and it clearly demonstrates that this is a pervasive feature of such markets. This explains the interest in both the DA and TTC mechanisms, since there is no matching mechanism

a	b	c
i	j	j
k	i	i
j	k	k

(a) Objects' Priorities

i	j	k
b	a	a
a	b	b
c	c	c

(b) Agents' Preferences

objects	a	b	c
agents	i	j	k

(c) Unique fair allocation

objects	a	b	c
agents	j	i	k

(d) Pareto-dominating allocation

Table 1: Incompatibility of Fairness and Pareto-efficiency

which is both fair and efficient in one-sided matching models. This fundamental incompatibility has inspired a substantial literature using various approaches to reconcile the properties of fairness and efficiency in some satisfying manner, and to understand and potentially alleviate the trade-offs involved.

Among these efforts, Heo (2019 [17]) restricts the preference domain and identifies restrictions that allow for both fairness and efficiency. More importantly for our analysis, Ergin (2002 [13]) and Kesten (2006 [21]) focus on restrictions on a priority table. Other papers try to weaken the notion of fairness or efficiency to allow a matching rule to satisfy both properties simultaneously without having to resort to restricting either preferences or priorities. Weakening the notion of fairness is especially popular and has created a large recent literature (see, for example, Alcalde and Romero-Medina (2017 [4]), Cantala and Pápai (2014 [8]), Morrill (2015 [27]), and Kloosterman and Troyan (2016 [23]), among others).

Another strand of the literature is primarily looking for alternative mechanisms that satisfy some weaker criteria than the original incompatible properties. For example, Kesten (2010 [22]) proposes a mechanism that is meant to improve stability compared to the TTC, based on consenting agents who allow for priority violations in order to improve the efficiency of the DA outcome. While this rule, called EADAM, maintains the efficiency of TTC, it is not strategy-proof. For many-to-one problems known as *school choice*, Morrill(2013 [26]) and Hakimov and Kesten (2014 [15]) introduce different matching rules that remain Pareto-efficient and strategy-proof, just like the TTC, but try to improve upon TTC by having fewer justified envy instances at some priority and preference profiles, but as we will show in Chapter 4, they can create more justified envy instances than TTC at some profiles, so the theoretical profile-by-profile comparison remains ambiguous.

As already discussed, strategy-proofness is not just an incentive property to assure truth-telling itself. The main reason to insist on strategy-proofness is that if a designer does not know the true rankings of agents who are participating in the market, then it is not possible to make sure that fairness and efficiency truly hold. In this thesis, our primary criteria for matching rules are efficiency and strategy-proofness, and we ask that if we maintain these two important properties of matching rules, is it possible to find a rule (or rules) that always perform better, in terms of fairness, than others, taking into account all different markets, or at least perform better on the whole? Thus, our studies fit into the literature analyzing the incompatibility of fairness and efficiency, and uses fairness comparison criteria to identify matching rules that are most suitable to reduce the fundamental trade-off between fairness and efficiency when efficiency takes precedence over fairness, while ensuring proper incentives and thus truthfully reported preferences.

There are different fairness criteria that can be used for comparison. Abdulkadiroğlu et al. (2017 [3]) introduce the notion of *justified envy minimality*. A matching rule is justified envy minimal if the set of justified envy instances cannot be weakly decreased in an inclusion sense by another rule, considering each priority and preference profile. Chen and Kesten (2017

[9]) use a different concept, which simply compares whether a profile contains a justified envy instance or not, and uses this criterion for comparison profile-by-profile. We call this criterion *JE-domination,* and this is the central concept studied in Chapter 2.

1.4 The Role of Simulations

Simulation is used as a complementary tool to our analytical methods, since some properties are not theoretically true for all possible profiles, but they are statistically significant. Some instances may happen in matching mechanisms which are rare or can be considered exceptions. Also, conjectures can be verified through simulations and properties that are found via simulations may result in a theoretical insight.

A case in point is Ashlagi et al. (2011 [5][6]) who ran simulations on kidney exchange to show how we should organize the chain of donors to obtain more efficient allocations. Roth and Peranson (1999 [33]) used simulations in school choice to see the effect of manipulation by agents in a large market. Dur et al. (2018 [10]) carried out simulations to compare the DA rule to an improved version of the mechanism that was used in Boston. Their simulations suggested that the Secure Boston mechanism Pareto-dominates the DA when the DA is not efficient, and it often overlaps with the DA when the DA is efficient. Miralles (2009 [25]) also ran simulations and proved that the Boston mechanism outperforms the DA when comparing the welfare of the two mechanisms. Morrill (2013 [26]) used simulations to show that Clinch and Trade (CT) and another variant of the TTC called Prioritized Top Trading Cycles (PTC), have fewer unfair instances than the TTC on average.

In Chapters 3 and 4 of this thesis we test different properties of matching rules in cases where analytical comparisons are not possible. In Chapter 3 simulations are used to show that more cycles in the priority profile lead to more fairness violations on average when considering different preferences for the agents. In Chapter 4, simulations are carried out to compare alternative matching mechanisms for the school choice model and see which one has a relative advantage over its rivals according to various normative criteria regarding fairness and efficiency.

1.5 Summary of the Results

This thesis consists of three studies in matching theory and market design. Its main focus is to compare matching rules according to normative criteria, primarily fairness, when objects have priorities over agents, in addition to agents having preferences over objects.

In the first study (Chapter 2), we analyze one-to-one matching, for which the two definitions of cycles in priority profiles introduced by Ergin (2002 [13]) and Kesten (2006 [21]) are identical. The fairness comparison criteria, called *JE-domination* (where JE stands for justified envy) and *cardinal JE-domination* that we focus on in this chapter are based on profile-by-profile comparisons. Given a fixed priority profile, a rule f JE-dominates another rule g at this priority profile if at each preference profile where g is fair

f is also fair, and if there is at least one more preference profile at which f is fair and g is not. The alternative notion, cardinal JE-domination, compares the number of justified envy instances of two matching rules. We say that rule f cardinally JE-dominates rule g at a fixed priority profile if there is no preference profile where f has more justified envy instances than g and there is at least one preference profile at which f has strictly fewer justified envy instances than g. We use these two straightforward criteria to compare strategy-proof and efficient matching rules. As we will show in Chapter 2, there is no unique rule that performs best at all possible preference profiles when the priority profile has at least one Ergin/Kesten cycle. Namely, we prove that when there is a cycle in the market, we cannot find a mechanism which is strategy-proof and Pareto-efficient and has weakly fewer preference profiles at which the allocation is fair than any other strategy-proof and Pareto-efficient mechanism, when the comparison is made profile by profile. This means that there is no unique rule at cyclic priority profiles which JE-dominates all other strategy-proof and efficient rules, and a similar result applies to cardinal JE-domination.

Gale and Shapley (1962 [14]) proves that there is a unique mechanism, the DA, that is fair and Pareto-dominates all other fair mechanisms (i.e., weakly Pareto-dominates all fair outcomes at each priority and preference profile). Some recent papers such as Morrill (2015) [27] and Abdulkadiroğlu et al. (2017 [3]) suggest that the TTC is more fair than any other mechanism that is strategy-proof and Pareto-efficient. Many papers have tried to reconcile Pareto-efficiency and fairness by either weakening the notion of fairness (e.g., Alcalde and Romero-Medina, 2017 [4]; Cantala and Pápai, 2014 [8]; Kloosterman and Troyan, 2016 [23]) or by restricting the priority profile (Ergin, 2002 [13]; Kesten, 2006 [21]). We show that while the TTC cannot be dominated in terms of fairness violations at all preference profiles among strategy-proof and Pareto-efficient mechanisms, if the priority profile is strongly cyclic then there is not much scope for the TTC to dominate other strategy-proof and Pareto-efficient matching rules.

In the second study (Chapter 3), we take a closer look at definitions that have been introduced by Ergin (2002 [13]) and Kesten (2006 [21]) in a many-to-one matching model. Given that this study focuses on multiple copies of objects, the scarcity conditions are important and the two definitions of cycles are no longer identical. We show, using a direct proof, that an Ergin cycle is stronger than a Kesten cycle when objects have quotas. Moreover, we also prove that when there is a Kesten cycle, which is the weaker definition of cycles, there is no strategy-proof and Pareto-efficient mechanism which has weakly fewer fairness violations (called *justified envy instances*) at all possible preference profiles compared to other strategy-proof and Pareto-efficient mechanisms. In the last part of Chapter 3 the matching prescribed by the TTC mechanism using different priority profiles is compared while we count the number of Kesten cycles in the priorities. It is shown that, as a general pattern, as the number of Kesten cycles increases, there are more justified envy instances and fewer preference profiles at which the mechanism produces fair allocations.

The third study (Chapter 4), which consists of multiple parts, explores three competing mechanisms for matching markets with multiple quotas, all of which are strategy-proof and Pareto-efficient but not fair. The TTC mechanism as a baseline is compared to the Equitable Top Trading Cycles mechanism (henceforth ETTC), and the Clinch and Trade mechanism (henceforth CT), as two alternative mechanisms, which were introduced as variations of TTC by Hakimov and Kesten (2018 [15]) and Morill (2015 [28]), respectively.

First, knowing that mechanisms that are efficient cannot be fair at all preference profiles, we compare these mechanisms that have been introduced in the literature for many-to-one matching in terms of fairness. Although one would expect that these two recently proposed strategy-proof and Pareto-efficient mechanisms are fairer than the TTC, I demonstrate that some preference profiles give the opposite result. Given that these alternative mechanisms are not always better in terms of fewer justified envy instances, I compare the aggregate number of justified envy instances using simulations, in order to see which mechanism has fewer justified envy instances than the others. The aggregate number is simply the sum of all justified envy instances in the selected priority and preference profiles.

Since all the alternative mechanisms that we study in this chapter are the same when the quota of all objects is equal to one, we compare the mechanisms when the the quota of at least one of the objects is greater than one. In order to be able to compare the inequality of different distributions of quotas over objects, we use the Gini coefficient. This coefficient is usually applied to the distribution of wealth, but we use it as a measure of inequality for distributions of quotas in matching (e.g., distributions of school seats across schools).

Moreover, there is another feature that has been used in the setup for our simulation model: namely, the preference profiles of agents are not completely random. They are a combination of a given common ranking and individual rankings over objects. We assume that there is a correlation between the common ranking and the individual preferences. The common ranking can be interpreted as common information for all the individuals, while individual rankings represent each individual's taste for the different objects. We carry out the analysis with different distributions of quotas and different correlations of preferences and show, among other things, that ETTC tends to have fewer aggregate justified envy instances than the other two rules.

Furthermore, by applying a new measure of efficiency, we show that each of the three mechanisms is more efficient when the most preferred object in the common ranking has a higher quota in general. Considering different distributions of quotas, we find that the more unequal the distribution, the less justified envy instances are observed in the market, and at the same time the more likely it becomes that agents are matched to objects that are ranked highly by them.

In sum, this thesis shows that in one-to-one matching there is no guarantee that that the TTC mechanism is the most fair mechanism when using some fundamental and straightforward criteria for comparison. Moreover, in many-to-one matching (i.e., with multiple quotas) the mechanisms cannot be compared easily analytically, especially when different quota distributions are considered. Therefore, for these cases simulations are used to compare mechanisms.

Chapter 2

Fairness Comparisons of Strategy-Proof and Efficient Matching Rules

2.1 Introduction

We study a one-to-one matching model where agents have strict preferences over objects and objects have strict priorities over agents. Our main question is which strategy-proof and efficient matching rules are the most fair? Can we compare these rules based on justified envy?

The set of strategyproof and efficient matching rules in this model has not been characterized yet. Pápai (2000 [29]) characterizes a large class of matching rules, called Hierarchical Exchange rules, using strategy-proofness, efficiency, nonbossiness (agents cannot affect other agents' allocations without affecting their own allocation), and reallocation-proofness (agents cannot manipulate the outcome successfully by reporting untruthful preferences and swapping their assigned objects afterwards). This class is a generalization of Gale's Top Trading Cycles (TTC) rule (Shapley and Scarf, 1962 [37]) and allows for an arbitrary priority profile that may change endogenously with the preference profile as the rounds of trading proceeds, based on prior assignments. Pycia and Ünver (2017 [30]) characterize a superset of these rules, called Trading Cycles rules, by strategy-proofness, efficiency, and nonbossiness. Trading Cycles rules allow for so-called brokers under restricted circumstances, where a broker is an agent who can trade her object but cannot take it herself. We study the even larger family of rules which are strategy-proof and efficient but may be bossy.

We examine two natural criteria that allow for comparing matching rules in terms of justified envy. We say that a rule *JE-dominates* another one if the first rule has no justified envy at any preference profile where the second one has no justified envy, and there is at least one additional preference profile at which the dominating rule has no justified envy, while the dominated one has. Furthermore, a rule *cardinally JE-dominates* another one if the number of justified envy instances are either the same or lower at each preference profile for the dominating rule, with at least one profile with a strictly lower number of justified envy instances.

It is well known that the Deferred Acceptance (DA) rule of Gale and Shapley (1962 [14]) is fair (in the sense that it has no justified envy) in this model but not efficient, while the TTC rule is efficient but not fair. Furthermore, we know that the DA outcome is the most efficient fair matching at each preference profile, in the strong sense that it Pareto-dominates every other fair matching. By symmetry, one might conjecture that the major competitor to DA, the Top Trading Cycle (TTC) rule, is the most fair (i.e., has least justified envy) among efficient rules. Along these lines, Morrill (2015 [27]) demonstrates that the TTC rule is the unique strategy-proof, efficient and *just* matching rule, where the axiom of justness weakens the notion of fairness to accommodate the trading of objects. Furthermore, Abdulkadiroğlu et al. (2017 [3]) shows that the TTC rule is justified-envy minimal among strategy-proof and efficient matching rules, in terms of an inclusion relation with respect to blocking pairs. However, the TTC rule is not the only matching rule with this property. Finally, although less closely related to our setting, there are also some arguments in favor of TTC on the basis of fairness when there is uncertainty: Harless (2015 [16]) shows that among ex-post efficient, strategy-proof, and nonbossy rules, TTC rules Lorenz dominate non-TTC rules.

We argue and demonstrate that when the object priorities are not acyclic (Ergin, 2002 [13]; Kesten, 2006 [21]) the TTC rule does not stand out as the unique most fair strategy-proof and efficient matching rule, contrary to the indications in the literature. Specifically, we show that with respect to either JE-domination or cardinal JE-domination there is no unique strategyproof and efficient rule that dominates all other such rules for arbitrary cyclic priorities. We further explore JE-domination among the class of Modified TTC rules, which are TTC rules that use a priority profile other than the "true" priority profile. The Modified TTC rules are also strategy-proof and efficient, and we prove that when the priorities are strongly cyclic (a strengthening of the classical cyclic condition) the TTC rule JE-dominates a Modified TTC rule only under rather restrictive conditions on the true priority profile.

2.2 Definitions and Axioms

There is a set of m objects M and a set of n agents N. Each agent is allocated at most one object and each object is assigned to at most one agent, based on the preferences of the agents and the priorities of the objects. Each object $a \in M$ has a strict priority ranking π_a of agents. Each agent $i \in N$ has a strict preference ordering P_i over objects. Strict preference relations are denoted by P_i and weak preference relations are denoted by R_i. Given that preferences are strict, aR_ib means that either aP_ib or $a = b$, that is, since indifference is not allowed if a is only weakly preferred to b but not strictly, then objects a and b are identical. Objects may be unacceptable to agents, and we denote the assignment of an agent who remains unassigned by 0. We also use the notation $P_i \in (a, b)$ to indicate that P_i ranks a first and b second, while $P_j = (a, 0)$ indicates that j's only acceptable object is a. Similarly, $P_i = (0)$ means that there is no acceptable object for agent i. We will use similar notation for priorities π_a. A *preference profile* $P = (P_1, \ldots, P_n)$ specifies the strict preferences of each agent in a particular market. A *priority profile* $\pi = (\pi_{a_1}, \ldots, \pi_{a_m})$ specifies the strict priority ordering of each object. Let Π denote the set of priority profiles.

A *matching* μ is a mapping from the set of agents N to the set of objects M such that each agent is assigned to at most one object, and each object is assigned to at most one agent. Thus, if agent i is assigned object a then $\mu(i) = a$. A *matching rule* f assigns a matching to each priority and preference profile (π, P). When the priorities are fixed and unambiguous, we may simply write $f(P)$ to indicate the matching at preference profile P. Agent i's assignment in $f(P)$ is denoted by $f_i(P)$.

A matching rule f is **strategy-proof** if, for all preference profiles P, there is no agent $i \in N$ and P'_i such that $f_i(P'_i, P_{-i})P_i f_i(P)$. If there is such an agent, then agent i can **manipulate** at profile P via P'_i. Strategy-proofness is a standard, although demanding, incentive requirement in matching design, and thus we will require all matching rules that we analyze in this study to satisfy it.

A matching rule f is **nonbossy** if, for all preference profiles P, there is no agent $i \in N$ and P'_i such that $f_i(P'_i, P_{-i}) = f_i(P)$ and $f(P'_i, P_{-i}) \neq f(P)$. If there is such an agent, then agent i is **bossy** at profile P, since she can change somebody else's assignment without changing her own. We will not require nonbossiness in this study, but will use this property of rules in our proofs, since for example the TTC rule is nonbossy.

Another basic property of matching rules is Pareto-efficiency, or efficiency for short. A matching μ is **individually rational** if for all $i \in N$, $\mu_i R_i 0$. A matching μ is **efficient** if it is not Pareto-dominated. That is, there is no matching ν such that, for all agents $i \in N$, $\nu_i R_i \mu_i$ and $\nu_j P_j \mu_j$ for some

$j \in N$. We assume that agents can always be left unassigned, and thus an efficient matching is individually rational. A matching rule f is **efficient** if, for all preference profiles P, $f(P)$ is efficient.

Finally, we introduce the central notions in this study on fairness which are also standard, based on the justified envy of agents with respect to their priority rankings for objects. Given a fixed priority profile π, a matching μ has **justified envy** at preference profile P if there exist agents $i, j \in N$ and object $a \in M$ such that $aP_i f_i(P)$, $i\pi_a j$, and $f_j(P) = a$. We will also use the terminology that (i, a, j) is a justified envy instance, or **JE instance** for short.[1] A matching rule f is **fair** if $f(P)$ has no justified envy at any preference profile P. We will say that g has JE at P (with respect to π) if $g(P)$ has at least one justified envy instance at P, and that g is JE-free at P if $g(P)$ has no justified envy instance at P (with respect to π).

2.3 Preliminaries

Here is a brief informal description of the TTC rule with inheritance of the objects for a given priority profile π. For a formal definition see the Fixed Endowment Hierarchical Exchange rules in Pápai (2000 [29]), and for a more succinct description of the ownership rights (i.e., endowments) see Pycia and Ünver (2017 [30]). The only, fairly straightforward, modification we make to previous definitions is that we allow agents to have unacceptable objects, and thus we remove agents from the market when all

[1] Note that in the one-to-one matching model that we study in this chapter the number of JE instances corresponds to the number of blocking pairs. This would no longer be the case in a many-to-one model, where an object has multiple copies which can be assigned to different agents.

their acceptable objects have already been assigned to other agents.[2]

- In each round each object is endowed to the agent who has the highest priority for the object among the agents who are still in the market.
- Each agent points to the agent who is endowed with the object that is most preferred by the agent among all the objects that are still unassigned. Agents may point to themselves.
- Since there is a finite number of agents, there is at least one cycle (i.e., a *top trading cycle*). Assign to each agent in each top trading cycle the object that the agent most prefers (that is, the object that the agent is pointing for). These assignment are final.
- Remove each assigned agent and object from the market, and remove each agent who has no acceptable objects left in the market.
- Update the endowments by endowing each object to the highest-priority agent who is still in the market. This is the "inheritance" of objects that are left behind by agents who leave the market in this round.
- Repeat this process in the remaining market until no more assignments can be made.

Given a permutation of the agents, the **Serial Dictatorship** rule assigns each agent her favorite object among the remaining objects, when they select their assignments in the order of the given permutation. Note that

[2] Pycia and Ünver (2016 [31]) explore this modification, what they call the existence of an outside option, in some detail.

if the fixed priority profile is homogeneous (has the same priority ordering for all objects), then the TTC rule is simply the Serial Dictatorship with the permutation of the agents that corresponds to the common priority ordering of the objects.

The TTC rule is strategy-proof and efficient (Pápai, 2000 [29]). Given an arbitrary priority profile, the TTC rule is not fair at all preference profiles. In fact, fairness and efficiency cannot be reconciled for an arbitrary priority profile, since there may be preference profiles for which a fair and efficient matching does not exist (Roth, 1982 [32]). When restricting the priority profile, fairness and efficiency can only be satisfied simultaneously if the priority profile is acyclic (Ergin, 2002 [13]; Kesten, 2006 [21]).

In a one-to-one matching model, which is the focus of this chapter, a priority profile is **acyclic** according to Ergin (2002 [13]) if there is no cycle for two objects $a, b \in M$ such that there are three agents $i, j, l \in N$ with $l \pi_a j \pi_a i$ and $i \pi_b l$. This turns out to be equivalent to the acyclicity condition defined by Kesten (2006 [21]), which also stipulates in addition that $i \pi_b j$. Given the symmetry between objects a and b in these definitions, it is straightforward to verify that the two conditions are the same (see details in the appendix).[3] Moreover, when the priority profile is acyclic, the TTC and DA rules are equivalent in the sense that both yield the same matching at each preference profile.

Since acyclicity of a priority profile is central to our results, we will state

[3]The two acyclicity conditions however differ from each other when objects have multiple copies or a quota, that is, in a many-to-one model. When defining the acyclicity of a priority profile, both Ergin (2002 [13]) and Kesten (2006 [21]) also specify a *scarcity* condition in addition to the *loop* condition to define a cycle. These scarcity conditions are trivially satisfied when the quota for each object is one, as assumed in this chapter.

the definition here formally.

Definition 1. A priority profile π has a **cycle** if there exist agents $i, j, l \in N$ and objects $a, b \in M$ such that $l\pi_a j\pi_a i$, $i\pi_b l$ and $i\pi_b j$. A priority profile is **cyclic** if it has a cycle, and it is **acyclic** if it does not have a cycle.

2.4 Fairness Criteria and Comparisons

Our first result demonstrates that none of the strategy-proof and efficient matching rules stands out uniquely in terms of a profile-by-profile fairness comparison when the priority profile is cyclic.

Theorem 1. *Let $n, m \geq 3$ and fix a priority profile π which is cyclic. Let matching rule f be strategy-proof and efficient. Then there exists another strategy-proof and efficient matching rule g and a preference profile P such that f has justified envy at P with respect to π and g does not.*

Proof. Since π is cyclic, there exist objects $a, b \in M$ and agents $a, b, c \in N$ such that $i\pi_a l\pi_a j$ and $j\pi_b i$. Assume, without loss of generality, that $j\pi_b l$. Consider a profile P such that $P_i, P_j, P_l \in (a, b)$ and for all other agents $k(k \in N \setminus \{i, j, l\}), P_k \in (k)$. If $f_{(i,j,l)}(P) \neq (a, b, 0)$, where this notation means that i gets a, j gets b and l gets 0, then let g be the TTC rule based on π. Then g is strategy-proof and efficient and it selects $(a, b, 0)$ at P, that is, $g_{(i,j,l)}(P) = (a, b, 0)$. Hence, g is JE-free at P. Given that f is efficient, it assigns both a and b at P. If $f_i(P) = a$ then $f_l(P) = b$ by efficiency. Furthermore, (j, b, l) is a JE instance for f at P. If $f_i(P) \neq a$ then efficiency implies that either $f_j(P) = a$ or $f_l(P) = a$. Thus, either (i, a, j) or (i, a, l)

is a JE instance for f at P. Therefore, if $f_{(i,j,l)}(P) \neq (a,b,0)$ then the statement follows for f, given the TTC rule based on π.

Next, assume that $f_{(i,j,l)}(P) = (a,b,0)$. Let $P'_i \in (b,a)$, and consider the profile $P' = (P'_i, P_{-i})$. Note that $f_i(P') \neq a$ by efficiency, and thus, given $f_i(P) = a$, strategy-proofness implies that $f_i(P') = b$. Then efficiency implies that $f_{(i,j,l)}(P') \in \{(b,a,0),(b,0,a)\}$.

Case 1: $f_{(i,j,l)}(P') = (b,a,0)$

Let $\hat{\pi}$ be such that $i\hat{\pi}_a l\hat{\pi}_a j$ and $i\hat{\pi}_b l\hat{\pi}_b j$. Let g be the TTC rule based on $\hat{\pi}$. For simplicity, we can assume that g is the Serial Dictatorship with permutation (i,l,j). Thus, g is strategy-proof and efficient.

Let $P''_j \in (a,b,c)$, where $c \in M$. Consider the profile $P'' = (P''_j, P'_{-j})$. Then $g_{(i,j,l)}(P'') = (b,c,a)$. Hence, g is JE-free at P'' with respect to π. However, note that since $f_j(P') = a$, strategy-proofness implies that $f_j(P'') = a$. This means that (l,a,j) is a JE instance for f at P'' with respect to π. Thus, the statement follows for f, given the Serial Dictatorship g with permutation (i,l,j) and profile P''.

Case 2: $f_{(i,j,l)}(P') = (b,0,a)$

Let $\bar{P}_j \in (b,a)$ and consider the profile $\bar{P} = (\bar{P}_j, P'_{-j})$. Since $f_b(P') = 0$, strategy-proofness implies that $f_b\bar{P} = 0$. Then by efficiency $f_{(i,j,l)}(\bar{P}) = (b,0,a)$. Now let g be the TTC rule based on π. Then g is strategy-proof and efficient and $g_{(i,j,l)}(\bar{P}) = (a,b,0)$. Note that g is JE-free at $\bar{P}$ with respect to π. However, (j,b,i) is a JE instance for f at $\bar{P}$ with respect to π. Thus, the statement follows for f, given the TTC rule g based on π and profile $\bar{P}$.

Since the statement is proved for both cases, the proof is complete. $\square$

Corollary 2. *Let $n, m \geq 3$ and fix a priority profile π which is cyclic. Let matching rule f be strategy-proof and efficient. Then there exists another strategy-proof and efficient matching rule g and a preference profile P such that f has more JE instances at P with respect to π than g.*

Corollary 2 follows immediately from Theorem 1.

Next we introduce the key definitions of this paper, the criteria which will allow us to evaluate and compare matching rules based on their fairness.

Definition 2. A rule f is **JE-dominated** if there exists another rule g which is JE-free at all preference profiles where f is JE-free, and g is JE-free at least at one preference profile where f has JE. Then we will say that g JE-dominates f.

Definition 3. A rule f is **cardinally JE-dominated** if there exists another rule g which has at most as many JE instances at each preference profile as f, and it has fewer JE instances than f at least at one preference profile. Then we will say that g cardinally JE-dominates f.

We note that, although closely related, there is no logical implication between JE-domination and cardinal JE-domination, which can be easily verified. JE-domination was first proposed by Chen and Kesten (2017 [9]).

Corollary 3. *Let $n, m \geq 3$. If π is cyclic then there is no unique strategyproof and efficient rule which JE-dominates all other strategyproof and efficient rules.*

Corollary 4. *Let $n, m \geq 3$. If π is cyclic then there is no unique strategy-proof and efficient rule which cardinally JE-dominates all strategy-proof and efficient rules.*

Corollaries 3 and 4 are straightforward implications of Theorem 1, using our new terminology.

2.5 JE-Domination for the TTC Rule

Given the priority profile $\pi \in \Pi$, we will denote the TTC rule based on π by TTC(π) and the DA rule based on π by DA(π).

Proposition 1. *If π is acyclic, the unique strategyproof and efficient rule which both JE-dominates and cardinally JE-dominates all other strategyproof and efficient rules is the $TTC(\pi) = DA(\pi)$ rule.*

Proof. It is well-known and easy to verify that the TTC and DA rules are equivalent if π is acyclic (Kesten, 2006 [21]), and therefore this rule is strategy-proof, efficient and fair. Moreover, the efficient and fair outcome chosen by this rule is the unique efficient and fair outcome at each preference profile. Suppose, to the contrary, that there is another efficient and fair outcome at some preference profile. Then neither outcome Pareto-dominates the other, since both are efficient. However, this contradicts the fact that the DA outcome is the agent-optimal fair outcome and thus it Pareto-dominates all other fair outcomes (Gale and Shapley, 1962 [14]). Therefore, the TTC rule based on π (or, equivalently, the DA rule based on π) chooses the unique efficient and fair matching at each preference profile, and it follows that any other efficient rule chooses an outcome at some preference profile which is not fair. $\square$

As Corollaries 3 and 4 indicate, Proposition 1 cannot be extended to priority profiles that are not acyclic. We now provide a simple example to demonstrate that, given a cyclic priority profile π, the TTC(π) rule neither JE-dominates nor cardinally JE-dominates all other strategy-proof and efficient rules.

Example 1. Let π be as follows: $\pi_a \in (j, i, l)$ and for all $b \in M \setminus \{a\}$, $\pi_b \in (i, l, j)$. Let f be the TTC rule based on π, and let g be the Serial Dictatorship based on agent permutation $(i, l, j, ...)$. Note that both rules are strategy-proof and efficient. Consider the preferences below, and assume that all other agents have no acceptable objects.

P_i	P_j	P_l
a	b	b
0	0	0

Then the TTC rule f based on π yields $f_i(P) = a$, $f_j(P) = b$ and f has a JE instance (l, b, j). However, the Serial Dictatorship g results in $g_i(P) = a$, $g_l(P) = b$, and g is JE-free at P.

Next we show that the TTC rule is not JE-dominated by any other strategy-proof and efficient rule, given an arbitrary priority profile $\bar{\pi}$.

Theorem 5. *Let $\bar{\pi} \in \Pi$ be a given priority profile. The TTC($\bar{\pi}$) rule is not JE-dominated by any other strategy-proof and efficient rule.*

Proof. Let g be the TTC($\bar{\pi}$) rule, given an arbitrary priority profile $\bar{\pi} \in \Pi$. Suppose, by contradiction, that there exists a strategy-proof and efficient rule f that JE-dominates g. Then there exists P such that $g(P)$ has JE and $f(P)$ is JE-free. Since $f(P)$ is efficient and JE-free, $f(P) = DA(P)$. Furthermore, $g(P) = \text{TTC}(P) \neq DA(P)$.

Since $g(P) = \text{TTC}(P)$ has JE, there exist $i, l \in N$ who trade at P in the TTC procedure and there exists $j \in N \setminus \{i, l\}$ who has justified envy at $g(P)$ for one of the objects, say $a \in M$, traded by i and l. This implies that if i and l trade $a, b \in M$ in the TTC procedure at P, then $i\bar{\pi}_a j\bar{\pi}_a l$ and $l\bar{\pi}_b i$, $l\bar{\pi}_b j$, (j, a, l) is a JE instance at $g(P)$, $g_i(P) = b$, $g_l(P) = a$, $P_i \in (b, \ldots)$ and $P_l \in (a, \ldots)$. Moreover, note that since $g(P)$ has JE, $g_j(P) = c$, where aP_jc and either $c \in M$ or $c = 0$.

Let $\overline{P}_i \in (b, a, 0)$, $\widetilde{P}_l \in (b, a, 0)$, $\widetilde{P}_i \in (a, b, 0)$ and $\overline{P}_l \in (a, b, 0)$. Suppose $f_i(\overline{P}_i, \overline{P}_l, P_j) = 0$. Then, by strategy-proofness, $f_i(\widetilde{P}_i, \overline{P}_l, P_j) = 0$. However, $g_i(\widetilde{P}_i, \overline{P}_l, P_j) = a$ and since the TTC rule is JE-free at $(\widetilde{P}_i, \overline{P}_l, P_j)$, $g(\widetilde{P}_i, \overline{P}_l, P_j) = f(\widetilde{P}_i, \overline{P}_l, P_j)$, which is contradiction. Thus, $f_i(\overline{P}_i, \overline{P}_l, P_j) \neq 0$. A similar argument for j shows that $f_l(\overline{P}_i, \overline{P}_l, P_j) \neq 0$.

Therefore, i and l are assigned by f objects a and b in total at $(\overline{P}_i, \overline{P}_l, P_j)$, and thus efficiency implies that $f_i(\overline{P}_i, \overline{P}_l, P_j) = b$ and $f_l(\overline{P}_i, \overline{P}_l, P_j) = a$. Then, by strategy-proofness, $f_i(\overline{P}_l, P_{-l}) = b$. Suppose $f_l(\overline{P}_l, P_{-l}) = a$. Then, by strategy-proofness, $f_l(P) = a$ would hold. We will show that this is a contradiction.

Suppose that $f_l(P) = a$. Then given that $f(P)$ is JE-free and $j\pi_a l$, there exists $d \in M \setminus \{a\}$ such that dP_ja and $f_j(P) = d$. If $d = b$, then $f_i(P) \neq b$.

Then, by strategy-proofness,$f_i(\overline{P}_i, P_{-i}) \neq b$.

Suppose that $f_i(\overline{P}_i, P_{-i}) \neq a$. Then $f_i(\overline{P}_i, P_{-i}) = 0$. Thus, by strategy-proofness, $f_i(\widetilde{P}_i, P_{-i}) = 0$. However , $g_i(\widetilde{P}_i, P_{-i}) = a$, and since the TTC rule is JE-free at $(\widetilde{P}_i, P_{-i})$, $g(\widetilde{P}_i, P_{-i}) = f(\widetilde{P}_i, P_{-i})$, which is a contradiction. Thus, $f_i(\overline{P}_i, P_{-i}) = a$. But then $f_l(\overline{P}_i, P_{-i}) \neq a$ and thus strategy-proofness implies that $f_l(\overline{P}_i, \overline{P}_l, P_j) \neq a$ which is a contradiction. Hence, $d \neq b$. However, this implies that matching (b, a, d) Pareto-dominates $g_{(i,l,j)} = (b, a, c)$, given that aP_jc and thus dP_jc. This is a contradiction, since the TTC rule is efficient. Therefore, $f_l(P) \neq a$ which contradicts our prior assumption.

Thus, $f_l(\overline{P}_l, P_{-l}) \neq a$. Hence, since $f_i(\overline{P}_l, P_{-l}) \neq b$, we have $f_l(\overline{P}_l, P_{-l}) = 0$. Then, by strategy-proofness, $f_l(\widetilde{P}_l, P_{-l}) = 0$. However, $g_l(\widetilde{P}_l, P_{-l}) = b$, and since the TTC rule is JE-free at $(\widetilde{P}_l, P_{-l})$, $g(\widetilde{P}_l, P_{-l}) = f(\widetilde{P}_l, P_{-l})$, which is a contradiction. $\square$

Note that Proposition 1 implies that only the DA rule and rules that coincide with the DA rule for acyclic priority profiles are not JE-dominated by strategy-proof and efficient matching rules for any given arbitrary priority profile, since the DA is fair for all priority profiles, and it is efficient for acyclic priority profiles.

2.6 JE-Domination for Modified TTC Rules

Since the characterization of all strategy-proof and efficient matching rules in the one-to-one model with priorities is an open problem, we will focus on the set of Modified TTC rules, which is a large class of well-studied matching rules (a subset of the Fixed Endowment Hierarchical Exchange rules of Pápai (2000 [29])) that are both strategy-proof and efficient. Let $\bar{\pi} \in \Pi$ be a given fixed priority profile. Then a **Modified TTC rule** is a TTC rule using priority profile $\pi \in \Pi \setminus \{\bar{\pi}\}$, denoted by TTC($\pi$), which does not use the given fixed priorities that we may refer to as the "true" priorities, but relies on some "modified" priorities as the basis for the TTC. This class of rules includes all Serial Dictatorships with different agent permutations, among many other rules.

First we provide a necessary condition for TTC($\bar{\pi}$) to JE-dominate a Modified TTC rule, where $\bar{\pi}$ is the "true" priority profile, and then we give a complete characterization of JE-domination by the TTC rule within this class of rules. Finally, we use these results to show that for strongly cyclic priority profiles, to be defined later, such JE-domination is not possible.

The intuition is straightforward for the necessity condition presented below. If TTC($\bar{\pi}$) JE-dominates TTC(π) then either the "true" priority profile $\bar{\pi}$ is acyclic, or if it is cyclic then for each cycle the *"intermediate"* agent (j) for an object, the agent whose priority is between the owner and recipient of an object, cannot be ranked higher than the two traders of the object, that is, both the owner (l) and the recipient (i).

Proposition 2. *Let the priority profile be $\bar{\pi} \in \Pi$. If TTC($\bar{\pi}$) JE-dominates TTC(π) for some $\pi \in \Pi \setminus \{\bar{\pi}\}$ then for each cycle given by $l\bar{\pi}_a j \bar{\pi}_a i$, $i\bar{\pi}_b l$, $i\bar{\pi}_b j$, we have either $l\pi_a j$ or $i\pi_a j$.*

Proof. Fix the priority profile $\bar{\pi} \in \Pi$. Let $\pi \in \Pi \setminus \bar{\pi}$ such that TTC($\bar{\pi}$)

JE-dominates TTC(π). Let f denote TTC($\bar{\pi}$) and let g denote TTC(π). Suppose that there is a cycle in $\bar{\pi}$, given by $l\bar{\pi}_a j\bar{\pi}_a i$, $i\bar{\pi}_b l$, $i\bar{\pi}_b j$, and suppose that $j\pi_a l$ and $j\pi_a i$. Let P be as follows: $P_i = (a, 0), P_j = (a, 0)$, $P_l = (b, 0)$ and for all $h \in N \setminus \{i, j, l\}$, $P_h = (0)$. Then i and l trade at P in TTC($\bar{\pi}$) and $f_i(P) = a$, $f_l(P) = b$, while $f_j(P) = 0$. However, $g_j(P) = a$. Then (j, a, i) is a JE instance at $f(P)$. Moreover, since $g_l(P) = b$ and $g_i(P) = 0$, by the efficiency of TTC, $g(P)$ is JE-free. This is a contradiction. □

Next we are going to present a characterization of JE-domination within this class of rules. Let $\bar{\pi} \in \Pi$ be the priority profile and let a cycle in $\bar{\pi}$ be given by $l\bar{\pi}_a j\bar{\pi}_a i$, $i\bar{\pi}_b l$, and $i\bar{\pi}_b j$. As before, we call j an *intermediate* agent for object a in this cycle, and we call the specific ways of modifying π defined below *intermediate transformation*, as it is centered on the role of the intermediate agent who could potentially have justified envy.

Definition 4. Priority profile $\pi \in \Pi$ is an **intermediate transformation** of $\overline{\pi}$ if for each cycle in $\overline{\pi}$ (as specified above) one of the following three cases holds:

case (i): $l\pi_a i$, $l\pi_a j$, $i\pi_b j$, $i\pi_b l$

case(ii): $i\pi_a j$, $i\pi_a l$

case (iii): $l\pi_a i\pi_a j$, $l\pi_b i$, $l\pi_b j$

To understand this definition, note that since j is an intermediate agent for object a, (j, a, i) is a JE instance if i is assigned to a. There are exactly three ways for i to be assigned to a if both agents i and j want object a and the third agent l wants a different object (see the preference profile

P_i	P_l	P_j
a	b	a

(a) preference profile

π_a	π_b
l	i
$i\,j$	$j\,l$

(b) **case (i):** i trades b for a

π_a	π_b
i	$\vdots$
$j\,l$	$\vdots$

(c) **case (ii):** i takes a

π_a	π_b
l	l
i	$i\,j$
j	

(d) **case (iii):** i inherits a

Table 2: Three cases for intermediate transformation

in Table 2), and these three ways correspond to the three cases in the definition, as illustrated by Table 2.

In the following we will consider the partitioning of each priority profile $\pi \in \Pi$, which is a partition of the "rows" of the priority profile when object priorities are listed vertically in a priority table, and each member of the partition consists of on or more consecutive rows in the table. That is, the priority profile is partitioned by consecutive rank numbers. Specifically, each agent belongs to exactly one member of this partition such that the agent only has priorities in this member of the partition. The partition that we are interested in is the finest partition of π which satisfies the above condition for each agent. We will refer to members of this uniquely defined finest partition as the *components* of π. The *size* of a component is given by the number of the agents in the component, which is simply the number of the rows in the priority table that correspond to the component. Thus, for example, each Serial Dictatorship has n components of size 1. It is also clear that acyclic priority profiles have components of size 1 and 2 only, while any cyclic priority profile has at least one size 3 component.

Now we are ready to prove the characterization of TTC JE-domination in the class of Modified TTC rules.

Theorem 6. *Let the priority profile be $\overline{\pi} \in \Pi$. For all $\pi \in \Pi \backslash \{\overline{\pi}\}$, $TTC(\overline{\pi})$ JE-dominates $TTC(\pi)$ if and only if π is an intermediate transformation of $\overline{\pi}$.*

Note that if $\overline{\pi}$ is acyclic then the conditions for intermediate transformation are vacuously satisfied due to the lack of cycles, and thus all priority profiles are intermediate transformations of $\overline{\pi}$. Therefore, one implication of this characterization is that all Modified TTC rules are JE-dominated by the TTC (other than TTC itself) if the priority profile has no cycles, and thus Theorem 6 partially implies Proposition 1.

Proof.

Claim 1: *If π is an intermediate transformation of $\overline{\pi}$ then $TTC(\overline{\pi})$ JE-dominates $TTC(\pi)$.*

Proof: Let π be an intermediate transformation of $\overline{\pi}$. Suppose that TTC($\overline{\pi}$) does not JE-dominate TTC(π). Then there exists $P \in \mathcal{P}$ such that TTC($\overline{\pi}, P$) has JE and TTC(π, P) is JE-free. Since TTC($\overline{\pi}, P$) has JE, there exist $a, b \in M$ and $i, j, l \in N$ such that $l\overline{\pi}_a j \overline{\pi}_a i$, $i\overline{\pi}_b l$, $i\overline{\pi}_b j$, $aP_i b$ and $bP_l a$, i and l will trade a and b at P and $aP_j f_j^{TTC}(\overline{\pi}, P)$. Thus, (j, a, i) is a JE instance for TTC($\overline{\pi}$) at P. The strategy-proofness and nonbossiness of TTC implies that without loss of generality we can assume that $P_i = (a)$, $P_l = (b)$ and $P_j = (a, f_j^{TTC}(\overline{\pi}, P))$. Assume also without loss of generality that the JE instance occurs occurs in the highest-ranked size 3 component in $\overline{\pi}$ for which such P exists.

Since the JE-free assignments are unique for all agents in the components before reaching the component containing $\{i, j, l\}$, their assignments are the same at TTC$(\overline{\pi}, P)$ and TTC(π, p). Moreover, if $\{i, j, l\}$ is not in the first component, then by our assumption there is no JE-instance including the agents in higher components at TTC$(\overline{\pi}, P)$, and since the JE-free assignments are unique, these agents get the same assignments at both TTC$(\overline{\pi}, P)$ and TTC(π, P). Repeating these arguments as many times as necessary, given the component structure of $\overline{\pi}$, we can conclude that a and b are not yet assigned at TTC(π, P) when we get to the component containing $\{i, j, l\}$, given that $f_i^{TTC}(\overline{\pi}, P) = a$ and $f_l^{TTC}(\overline{\pi}, P) = b$.

Now we can consider the three cases in the definition of intermediate transformation, given that π is an intermediate transformation of $\overline{\pi}$.

case (i): $l\pi_a i$, $l\pi_a j$, $i\pi_b j$, $i\pi_b l$.

In this case i and l trade a and b at TTC(π, P), and thus (j, a, i) is a JE instance at TTC(π, P)

case (ii): $i\pi_a j$, $i\pi_a l$.

Since $P_i = (a)$ and i ranks a first in π_a among the remaining agents, i is assigned a at TTC(π, P), and thus (j, a, i) is a JE instance at TTC(π, P)

case (iii): $l\pi_a i\pi_a j$, $l\pi_b i$,$l\pi_b j$

Since $P_l = (b)$ and l ranks first in π_b among the remaining agents, l is assigned b at TTC(π, P). Therefore, a is inherited by i from l, and since $P_i = (a)$, i is assigned a at TTC(π, P). Thus, (j, a, i) is a JE instance at TTC(π, P).

We have shown that TTC(π, P) has JE in all three cases, which is a contradiction. Hence, TTC$(\overline{\pi})$ JE-dominates TTC(π).

Claim 2: *If TTC($\overline{\pi}$) JE-dominates TTC(π) then π is an intermediate transformation of $\overline{\pi}$.*

Proof: Assume that $TTC(\overline{\pi})$ JE-dominates TTC(π). Suppose that π is not an intermediate transformation of $\overline{\pi}$, then there is a component of $\overline{\pi}$ for which there exists a cycle with $a, b \in M$ and $i, j, l \in N$ such that $l \overline{\pi}_a j \overline{\pi}_a i$, $i \overline{\pi}_b l$, $i \overline{\pi}_b j$ and none of the three cases in the definition of intermediate transformation holds for this cycle in π. Let P be such that $P_i = (a)$, $P_l = (b)$ and $P_j = (a)$, and for all other agents $k \in \setminus\{i, j, j\}$, $P_k = (0)$. Then i and l trade a and b at TTC$(\overline{\pi}, P)$, and thus (j, a, i) is a JE instance at TTC$(\overline{\pi}, P)$. Therefore, since TTC$(\overline{\pi})$ JE-dominates TTC(π), TTC(π, P) also has JE. This implies that j is not assigned a at TTC(π, P), since then l is assigned b at TTC(π, P) by Pareto-efficiency of the TTC, and TTC(π, P) would be JE-free, a contradiction.

If $j \pi_a i$ and $j \pi_a l$ then j is assigned a at TTC(π, P). Therefore, j is not ranked first by π_a among $\{i, l, j\}$, and since i is not ranked first by π_a among $\{i, l, j\}$, given case (ii) it must be the case that l is ranked first by π_a among $\{i, l, j\}$, that is, $l \pi_a i$ and $l \pi_a j$.

Case 1: $l \pi_a i \pi_a j$

Since i is not ranked first by π_b among $\{i, l, j\}$, given case (i), and l is not ranked first by π_b among $\{i, l, j\}$, given case (iii), it must be the case that j is ranked first by π_b among $\{i, l, j\}$, that is, $j\pi_b i$ and $j\pi_b l$. This implies that l and j trade a and b and thus j is assigned a at $\mathrm{TTC}(\pi, P)$.

Case 2: $l\pi_a j\pi_a i$

Since i is not ranked first by π_b among $\{i, l, j\}$, given case (i), either l or j is ranked first by π_b among $\{i, l, j\}$. If $l\pi_b i$ and $l\pi_b j$ then l is assigned b and j inherits a, and therefore j is assigned a at $\mathrm{TTC}(\pi, P)$. If $j\pi_b i$ and $j\pi_b l$ then l and j trade a and b, so j is again assigned a at $\mathrm{TTC}(\pi, P)$.

This shows that in each case j is assigned a at $\mathrm{TTC}(\pi, P)$ and we have reached a contradiction. Therefore, π is an intermediate transformation of $\overline{\pi}$. □

The relevance of the above results, the necessity condition in Proposition 2 and the characterization given in Theorem 6, is that in the presence of cycles in the priority profile, which are likely to occur in large numbers unless there is a reason for priorities to be nearly identical across objects, the TTC rule JE-dominates a Modified TTC rule only if the specified conditions are met by the modified priority table, which makes such JE-domination quite restricted. We will provide another result in this direction which will further illuminate the issue in the next section. This next theorem builds on the previous two results and provides more intuition about the restrictions and the limited nature of JE-domination of Modified TTC rules by the TTC rule.

2.7 Fairness Comparisons with Strong Cycles

Now we will explore JE-domination when the priority profile is not only cyclic but has some *strong cycles* as well. A strongly cycle consists of four agents and three objects which form a *double 4-cycle* in the priority profile. A *4-cycle* consists of a pair of priority orderings of four agents for two objects such that, given the priority ranking of the four agents for one object, the other object has the exact reverse priority ordering of the four agents. For $a \in M$, let the reverse of π_a be denoted by $\hat{\pi}_a$. If there are only for agents then a 4-cycle is given by π_a and π_b such that $\pi_b = \hat{\pi}_a$. Furthermore, a double 4-cycle is given by π_a, π_b and π_c such that $\pi_b = \pi_c = \hat{\pi}_a$. A general definition is provided below.

Definition 5. A priority profile π has a **strong cycle** if it has a double 4-cycle. Namely, there are agents $h, i, j, l \in N$ and objects $a, b, c \in M$ such that $l \pi_a j \pi_a i \pi_a h$, $h \pi_b i \pi_b j \pi_b l$ and $h \pi_c i \pi_c j \pi_c l$.

Note that a strong cycle in the priority profile implies that it is cyclic, but not necessarily the other way around. In particular, if the largest component of a priority profile is of size 3 then the priority profile is cyclic but does not have a strong cycle. If at least one component of a priority profile is at least of size 4 then it is cyclic, and it may also contain a strong cycle. However, no matter how large each component is, it is not guaranteed that the priority profile has strong cycles.

We now present a result which implies that the TTC rule does not, in general, JE-dominate Modified TTC rules. Although Theorem 6 is a characterization, we cannot easily infer from it the scope of the restrictions it places on JE-domination by the TTC. The next theorem sheds more light on these restrictions, as it shows that if the TTC rule were to JE-dominate a Modified TTC rule then the priorities involved in each strong cycle would have to be the same in the priority profile of the Modified TTC as the true priorities. We remark that for our purposes 4-cycles are as large as needed to establish our result, and no further generalization of cycles is required. This indicates that with a relatively small sized cycle, assuming that we have just two objects with exact reversals in priorities and thus have a strong cycle, we can already obtain a result that shows clearly that JE-domination by the TTC is rather limited.

We will use the following definition. Two priority profiles $\pi, \pi' \in \Pi$ are **in agreement** over a component, cycle, or strong cycle in π if the priorities for all involved objects are the same over all the involved agents in π' as in π.

Theorem 7. *Let the priority profile be $\overline{\pi} \in \Pi$. For all $\pi \in \Pi$, $TTC(\overline{\pi})$ JE-dominates $TTC(\pi)$ only if they are in agreement over each strong cycle.*

The proof of the theorem makes use of both Proposition 2 and Theorem 6.

Proof. Let the priority profile $\overline{\pi} \in \Pi$ have a strong cycle with $h, i, j, l \in M$ and $a, b, c \in M$ such that $l\pi_a j\pi_a i\pi_a h$, $h\pi_b i\pi_b j\pi_b l$ and $h\pi_c i\pi_c j\pi_c l$. Let $\pi \in \Pi$ such that TTC($\overline{\pi}$) JE-dominates TTC(π). We will show that then $\overline{\pi}$ and π are in agreement over this strong cycle.

Note first that there are four cycles in this strong cycle:

Cycle A_l : $l\overline{\pi}_a j\overline{\pi}_a i, i\overline{\pi}_b l, i\overline{\pi}_b j$

Cycle A_j : $j\overline{\pi}_a i\overline{\pi}_a h, h\overline{\pi}_b j, h\overline{\pi}_b i$

Cycle B_h : $h\overline{\pi}_b i\overline{\pi}_b j, j\overline{\pi}_a h, j\overline{\pi}_a i$

Cycle B_i : $i\overline{\pi}_b j\overline{\pi}_b l, l\overline{\pi}_a i, l\overline{\pi}_a j$

Based on cycle A_l, Proposition 2 implies that either $l\pi_a j$ or $i\pi_a j$.
Based on cycle A_j, Proposition 2 implies that either $j\pi_a i$ or $h\pi_a i$.
Therefore, if $i\pi_a j$ then $h\pi_a i$ and thus $h\pi_a i\pi_a j$, and if $j\pi_a i$ then $l\pi_a j$ and thus $l\pi_a j\pi_a i$. In sum, either $h\pi_a i\pi_a j$ or $l\pi_a j\pi_a i$.
We can show symmetrically for π_b, based on cycles B_h and B_i, that either $l\pi_b j\pi_b i$ or $h\pi_b i\pi_b j$.

Case 1: If $l\pi_a j\pi_a i$, then by Theorem 6 we have case (i) and $i\pi_b j, i\pi_b l$. Thus, $h\pi_b i\pi_b j$ and $h\pi_b i\pi_b l$ must hold. If we have $j\pi_a h\pi_a i$, then by Theorem 6 we have case (iii) and $j\pi_b i$ which is a contradiction. Therefore, $l\pi_a j\pi_a i\pi_a h$. We can show symmetrically that in this case $h\pi_b i\pi_b j\pi_b l$. We can also extend the above arguments to π_c to show that $h\pi_c i\pi_c j\pi_c l$ in this case. Hence, all priorities are in agreement in $\overline{\pi}$ and π over the four agents and objects a, b and c , which means that $\overline{\pi}$ and π are in agreement over this strong cycle.

Case 2: If $h\pi_a i\pi_a j$, then by Theorem 6 we have case (ii), and $l\pi_b j\pi_b i$ must also hold. We can show similarly that in this case $l\pi_c j\pi_c \pi_c i$ holds as well. Now let $P_i = P_j = (a, 0)$, $P_l = (b, 0)$ and $P_h = (c, 0)$. For all $k \in N \setminus \{h, i, j, l\}$, let $P_k = (0)$. Note that since object a is unacceptable to agent l and objects b and c are unacceptable to agent h, the priority ranking of l in π_a and the priority ranking of h in both π_b and π_c are irrelevant. Then $f^{TTC}_{(h,i,j,l)}(\overline{\pi}, P) = (c, a, 0, b)$ and TTC($\overline{\pi}$) has JE at P, since (j, a, i) is a JE instance at this profile. However, $f^{TTC}_{(h,i,j,l)}(\pi, P) = (c, 0, a, b)$ and TTC(π) is JE-free at P. This is a contradiction, given that TTC($\overline{\pi}$) JE-dominates TTC(π). $\square$

For any priority profile π, let the reverse of π be denoted by $\hat{\pi}$, where the reverse means the exact opposite ordering of agents for each object. Let the TTC rule based on π (the "true" priorities) be denoted by f and let the Reversed TTC rule based on the reversed priorities be denoted by $\hat{f}$. That is, $f =$ TTC(π) and $\hat{f} =$ TTC($\hat{\pi}$).

As noted above, the presence of a strong cycle requires that $n \geq 4$ and $m \geq 3$. We will now show that if we have either less than 4 agents or less than 3 objects then the TTC rule JE-dominates the Reversed TTC rule, given any arbitrary priority profile. What we show is that f JE-dominates $\hat{f}$ for very small markets, and it follows from Theorem 7 that JE-domination for any market that exceeds this size is only possible if the true priorities are preserved for agents and objects involved in a strong cycle. Thus, since the priority profile in a larger market may contain a strong cycle, this result cannot be extended to larger markets.

Proposition 3. *If $n < 4$ or $m < 3$ then the TTC(π) rule JE-dominates the Reversed TTC rule TTC($\hat{\pi}$) for an arbitrary priority profile $\pi \in \Pi$.*

Proof. First we note that if $n = 2$ or $m = 1$ then the priority profile is trivially acyclic and thus the result follows from Proposition 1. Therefore, we only need to prove the statement for $n = 3$ or $m = 2$.

Assuming that $n = 3$ and $m \geq 2$, or alternatively $m = 2$ and $n \geq 3$, it is easy to check that the only way the TTC rule has JE at some preference profile is if two agents trade in the first round of the TTC rule, and there is a third agent who does not receive an object in the first round and has justified envy either for one or both traded objects. In any other scenario the preference profile is JE-free for the TTC rule. We will show that in all of these cases the Reversed TTC rule has at least as many JE instances as the TTC rule. Let's assume that agent i has top priority for object b, agent j has top priority for object a, and they trade their objects in a top trading cycle in the first round of the TTC rule at a given preference profile P. Thus, (i, a) and (j, b) are both TTC assignments made in the first round of the procedure. Assume also that there is agent k who does not receive an object in the first round at P in the TTC rule, and has either one JE instance, (k, a, i), or two JE instances, (k, a, i) and (k, b, j). Let $(k, \tilde{c})$ be also an assignment, where $\tilde{c} \in (c, 0)$ and $c \in M$, that is, agent k may or may not be assigned an object in the second round of the TTC rule at the given profile. Note that in both cases, whether there is only one or two JE instances, we have $j\pi_a k\pi_a i$, $i\pi_b k$, and $i\pi_b j$. Also, $i\hat{\pi}_a k\hat{\pi}_a j$, so (i, a) is an assignment at P for the Reversed TTC rule.

Case 1: $i\pi_b k\pi_b j$

Then $j\hat{\pi}_b k\hat{\pi}_b i$. If (k, a, i) is the only JE instance at P for the TTC rule then $P_k \in (a, \tilde{c})$ and both the TTC and the Reversed TTC rules make the same assignments, (i, a), (j, b) and $(k, \tilde{c})$, so the Reversed TTC rule has the same JE instance. If both (k, a, i) and (k, b, i) are JE instances at P for the TTC rule then $P_k \in (a, b, \tilde{c})$ or $P_k \in (b, a, \tilde{c})$ and once more both the

TTC rule and the Reversed TTC rule make the same assignments, namely (i, a), (j, b) and $(k, \tilde{c})$, so the Reversed TTC rule has the same two JE instances as the TTC rule.

Case 2: $i\pi_b j\pi_b k$

Then (k, a, i) is the only JE instance at P for the TTC rule, and given that $k\hat{\pi}_b j\hat{\pi}_b i$, we have one of two cases. If either $P_k \in (a, b, \tilde{c})$ or $P_k \in (b, a, \tilde{c})$ then the Reversed TTC rule assigns (k, b) which implies that (j, b, k) is a JE instance for the Reversed TTC rule at P, so the Reversed TTC rule does not have fewer JE instances at P than the TTC rule. If $P_k \in (a, \tilde{c})$ then the Reversed TTC rule makes the same assignments as the TTC rule, (i, a), (j, b) and $(k, \tilde{c})$, so the Reversed TTC rule has the same JE instance as the TTC rule at P.

So far we proved that if TTC rule has JE at some preference profile then so is the Reversed TTC rule for all cases in the statement of the proposition, and in fact the Reversed TTC rule has at least as many JE instances at each preference profile as the TTC rule. Now we will show that there exists a preference profile $\bar{P}$ for each priority profile π such that TTC rule is JE-free at $\bar{P}$ given π, but the Reversed TTC rule has JE. This will prove that the TTC rule both JE-dominates and cardinally JE-dominates the Reversed TTC rule.

If π is acyclic then this (and the entire proposition) follows immediately from Proposition 1. Thus, we can assume that π is not acyclic, and then without loss of generality we have $j\pi_a k\pi_a i$, $i\pi_b k$ and $i\pi_b j$. Let $\bar{P}_k = (a, b, \tilde{c})$, and $\bar{P}_i = \bar{P}_j = (b, a, \tilde{c})$, where $\tilde{c} \in \{c, 0\}$ and $c \in M$. Then $f_{(i,j,k)}(\bar{P}) =$

$(b, a, \tilde{c})$ and the TTC rule is JE-free at this profile. Consider first $k\pi_b j$. Then $j\hat{\pi}_b k\hat{\pi}_b i$, which implies that $\hat{f}_j(\bar{P}) = b$. Then (i, b, j) is a JE instance for $\hat{f}$ at $\bar{P}$. Consider next $j\pi_b k$. Then $k\hat{\pi}_b j\hat{\pi}_b i$, and this together with $i\hat{\pi}_a k\hat{\pi}_a j$ implies that in the Reversed TTC rule agents i and k trade a and b. Thus, $\hat{f}_k(\bar{P}) = a$ and $\hat{f}_j(\bar{P}) = \tilde{c}$. Then (j, a, k) is a JE instance for the Reversed TTC rule $\hat{f}$ at $\bar{P}$. This concludes the proof. □

Finally, we remark that JE-domination is possible among Modified TTC rules even if the TTC rule does not JE-dominate any Modified TTC rule due to strong cycles in the priority profile. Namely, given an arbitrary true priority profile, if we select an acyclic priority profile which is consistent with the true priority orderings within each size 2 component of the priority profile, then the Modified TTC rule using this acyclic priority profile JE-dominates any other Modified TTC rule which is based on a priority profile with the same components.

2.8 Conclusion

In this chapter we showed that the TTC rule in general does not stand out among strategy-proof and efficient matching rules as the most fair rule with the least amount of justified envy based on profile-by-profile comparisons. When the priority profile is acyclic the TTC rule dominates all other strategy-proof and efficient rules, since it has no justified envy at any preference profile. However, in this case the TTC rule is equivalent to the DA rule, so the positive result is clear in this restricted case. We show that this is no longer true for the TTC rule if the priority profile is cyclic, since in this case there is no unique strategy-proof and efficient rule that dominates all the others when comparing the fairness of the rules at each preference profile. We characterize the set of priority profiles for which the TTC rule JE-dominates a Modified TTC rule, which gives us some idea about the restrictions for JE-domination when cycles are present in the priorities. This contrasts with the case of acyclic priority profiles, for which the TTC JE-dominates all other Modified TTC rules, which goes back to the equivalence of the DA and TTC in this relatively rare scenario.

We go one step further and also prove that when the priority profile has strong cycles a Modified TTC rule is only JE-dominated by the TTC rule if the modified priorities are in agreement with the true priorities for all strong cycles. This is the most restrictive case yet. Indeed, if there is just one object for which the priorities are completely reversed compared to all the other objects for which the priorities are in agreement, or if objects may only have a given priority ordering of agents or its exact opposite, just to mention a couple of obvious cases, then Theorem 7 implies that the TTC rule does not JE-dominate any of the Modified TTC rules, since in these scenarios all agents and objects are involved in a strong cycle. Given that markets are likely to have many strong cycles, especially since the cycle length is only 4 in strong cycles, unless there is a reason for the priorities across different objects to be highly correlated, this result demonstrates that the TTC rule is generally not fairer, based on a straightforward criterion of fairness, than many other known strategy-proof and efficient matching rules.

Appendix to Chapter 2

Ergin loop vs. Kesten loop:

Since scarcity conditions are trivially satisfied in a one-to-one model, we only need to consider the loop conditions to show that an Ergin cycle is the same as a Kesten cycle when each object has a quota of one, i.e., in the one-to-one matching model.

- **Ergin loop:** $l\pi_a j\pi_a i$ and $i\pi_b l$
- **Kesten loop:** $l\pi_a j\pi_a i$, $i\pi_b l$ and $i\pi_b j$

In the Ergin loop there are two possibilities:

1. $i\pi_b j$: This implies a Kesten loop immediately.
2. $j\pi_b i$: Then $j\pi_b i\pi_b l$, $l\pi_a j$ and $l\pi_a i$ is a Kesten loop.

Therefore, an Ergin loop always implies a Kesten loop. The other direction is immediate.

Chapter 3

Fairness Comparisons with Multiple Quotas

3.1 Introduction

Among the strategy-proof and Pareto-efficient one-sided matching rules when there are multiple quotas, we are looking for rules which are either stable or have a fewer number of justified envy instances when the priority table is cyclic than any other strategy-proof and Pareto-efficient matching rule. We review two different concepts of cycles, which are given by Ergin (2002 [13]) and Kesten (2006 [21]). First, we show that Ergin's cycle is stronger than Kesten's cycle. Second, we show that when there is a Kesten cycle,which is a weaker notion for cycles, there is no superior mechanism with the least number of justified envy instances at all possible preference profiles.

We study matching problems with agents and objects in which objects have priorities instead of preferences over agents. The difference is that object priorities can be assumed to be exogenous and fixed, and there is no strategic behavior on the object side. Furthermore, in one-sided matching, where there are agents on one side of the market only, the efficiency measures consider the agents only and not the objects.

In this chapter we assume that objects may have multiple copies, and thus we specify a quota for each object. This class of matching is known as many-to-one matching, a generalized form of one-to-one matching. In these models each agent can be matched only to one object, and the number of agents that are matched to an object cannot exceed the quota of the object. Given that here one side of the market has objects instead of agents, and the objects have fixed priorities over agents, this problem is also known in the literature as the *school choice problem* (Abdulkadiroğlu and Sönmez, 2003 [2]).

Here again an agent has envy for objects that the agent prefers compared to their currently allocated object. Furthermore, the envy is justified when the agent has higher priority than at least one of the agents who has been matched to the object. For this reason in one-sided matching theory we use the term of justified envy instead of a blocking pair that is used when we study two-sided matching with agents on both sides. If a mechanism eliminates all justified envy instances then it is fair. If a mechanism eliminates all blocking pairs then it is called stable. Thus, technically, the lack of justified envy, i.e., fairness, is identical to the lack of blocking pairs, i.e., stability in this context. We use the terminology of justified envy and fairness because this is the more appropriate terminology in school choice problems.

This study considers mechanisms that are strategy-proof and Pareto-efficient. The mechanism is strategy-proof if no agent can benefit by misreporting their true preferences. The mechanism is Pareto-efficient if we cannot improve the matching of any agent unless at least one of the agents is made worse off.

Gale and Shapley (1962 [14]) has an example which shows that the unique fair matching does not give any agent their first choice, implying that this matching is not Pareto-efficient for the agents. Balinski and Sönmez (1999 [7]) shows explicitly that a matching that satisfies both fairness and Pareto-efficiency may not exist at a particular preference profile. Kesten (2006 [21]) proves that a Pareto-efficient and strategy-proof rule cannot select the Pareto-efficient and fair allocation at each preference profile where it exists.

If the acyclicity property holds for priority tables, we can have both Pareto-efficiency and fairness simultaneously. The acyclicity property holds when we do not have cycles. A cycle is defined in two different ways by Ergin (2002 [13]) and Kesten (2006 [21]). We show that Kesten acylicity implies Ergin acyclicity or, equivalently, Ergin's cycle implies Kesten's cycle. This has been proved by Kesten (2006 [21]) by applying several different theorems. We provide an alternative direct proof.

We show that it is enough to have a Kesten's cycle, which is the weaker version of a cycle, to prove that there is no strategy-proof and Pareto-efficient rule which has weakly fewer justified- envy instances at all profiles than any other strategy-proof and Pareto-efficient rule.

3.2 Model

There are a finite number of agents $A = \{i, j, l, k, ...\}$ and a finite number of objects $O = \{a, b, c, ...\}$. Objects may have multiple copies and each object $ainO$ has a quota $q_a \geq 1$. The vector of quotas is denoted by $q = (q_a, q_b, q_c, \ldots)$. Each object has strict priorities $\succ_a$ over agents[1] and agents have preferences R_i over objects. aR_ib indicates that agent i weakly prefers a to b and $aP_i = b$ indicates that agent i strictly prefers a to b. A preference profile specifying a preference ordering for each agent is denoted by R, and the set of preference profiles is $\mathbb{R}$. We write $i \succ_a j$ if object a gives a strictly higher priority to agent i than to agent j. A priority profile specifying a priority ordering for each agent is denoted by $\succ$.

A profile of strict preferences of all agents is denoted by R and a profile of strict priorities of all objects is denoted by $\succ = (\succ_a)_{a \in O}$. $f_i(R) = a$ if agent i is matched to object a. If agent i stays unmatched we write $f_i(R) = i$. Given $a \in O$, $\succ'_a$and $i \in A$, let $U_i(a)$ be the set of agents who are preferred to i by object a. That is, $U_a(i) = \{j \in A | j \succ_a i\}$. Let q be a a quota vector such that element q_a is a quota associated with item a. For each agent $i \in N$ preference ordering R_i is a linear order of i over $O \cup \{i\}$.

[1]We use different notations here from the previous chapter, following some conventions used in the literature on school choice problems.

A matching μ is a function from agents to objects: $\mu : A \to O$ such that for each agent i and each object a if $\mu(i) = a$ then we can also write $i \in \mu^{-1}(a)$, and each object a is matched to no more than q_a agents, i.e., the number of agents matched to an object cannot exceed its quota: $|\mu^{-1}(a)| \leq q_a$. The set of all possible matchings is denoted by $\mathbb{M}$. Given a fixed priority profile, a mechanism or matching rule f is a function from all possible preference profiles to the set of matchings: $f : \mathbb{R} \to \mathbb{M}$.

A matching rule is *fair* if there does not exist any agent who has *justified envy* at any preference profile, given the fixed priority profile $\succ$. Agent i has justified envy for object a at a preference profile $\mathcal{R}$ such that $aP_i f_i(R)$, $i \succ_a j$ and $f_j(R) = a$. In this case student i's envy is justified.

A matching μ is *Pareto-efficient* at preference profile R if there is no matching ν such that at least for one agent i, $\nu(i)P_i\mu(i)$ and for all $j \in A$, $\nu_j R_j \mu_j$. for any other agent j. A matching rule is $Pareto-efficient$ if it assigns a Pareto-efficient matching to each preference profile.

A matching rule is *strategy-proof* if for all preference profiles R and agent i there is no R'_i such that $f_i(R'_i, R_{-i})P_i f_i(R)$.

3.3 Matching Rules

There are two matching rules studied in this chapter: Serial Dictatorships and Top Trading Cycles.

3.3.1 Serial Dictatorships

A Serial Dictatorship (SD) matching rule (Satterthwaite and Sonnenschein, 1981 [36]; Svensson, 1999 [38]) specifies an order of agents and lets the first agent receive her favorite object, then the next agent receives her favorite object among the remaining objects, and so on. The mechanism ends when there are no more agents in the list for whom there is an available object to be matched with or if each agent is already assigned an object.

3.3.2 Top Trading Cycles

A Top Trading Cycles (TTC) matching rule (Shapley and Scarf; 1962 [37]; Pápai, 2000 [29]; Abdulkadiroğlu and Sönmez, 2003 [2]) consists of the following steps:

- Each object forms a pair with its highest priority agent who is in the market.
- Each agent points to the pair with their most preferred object (which may be their own pair).
- Since there is a finite number of agents, there is going to be at least one cycle of pointing agents which, if carried out, improves the allocation of the involved agents (or leaves the allocation unchanged if an agent points to their own pair).
- Agents are permanently matched to the objects according to the cycles.

- Matched agents and one copy of each matched object is removed from the market and the process is repeated iteratively in the remaining market.
- The algorithm terminates when there are no more agents in the market who want to be matched to objects that have remaining copies in the market.

Both the Serial Dictatorships and the Top Trading Cycles rules are strategy-proof and Pareto-efficient.

3.4 Ergin versus Kesten cycles

Definition [Ergin, 2002 [13]]: Let $\succ$ be a priority profile and q a vector of quotas. An *Ergin cycle* consists of distinct $a, b \in O$ and $i, j, l \in A$ such that the following is satisfied:

- *Loop Condition*: $i \succ_a l \succ_a j$ and $j \succ_b i$.
- *Scarcity Condition*: There exist possibly empty *disjoint* sets of agents $N_a^e, N_b^e \subset N \setminus \{i, j, l\}$ such that $N_a^e \subset U_a(l)$ and $N_b^e \subset U_b(i)$, where $|N_a^e| = q_a - 1$ and $|N_b^e| = q_b - 1$.

A priority profile is *Ergin acyclic* if it has no Ergin cycle.

Definition [Kesten, 2006]: Let $\succ$ be a priority profile and q a vector of quotas. A *Kesten cycle* consists of distinct $a, b \in O$ and $i, j, l \in A$ such that the following is satisfied:

- *Loop Condition*: $i \succ_a l \succ_a j$, $j \succ_b i$ and $j \succ_b l$.
- *Scarcity Condition*: There exists a possibly empty set of agents $N_a^k \subset N \setminus \{i, j, k\}$ such that $N_a^k \subset U_a(i) \cup (U_a(l) \setminus U_b(j))$ and $|N_a^k| = q_a - 1$.

A priority profile is *Kesten acyclic* if it has no Kesten cycle. Note that if $q_a - 1 > 0$ then there has to be a set N_a^e satisfying the definition with $|N_a^e| = q_a - 1 > 0$, i.e., $N_a^e \neq \varnothing$, and otherwise the scarcity condition does not hold and the priority table is going to be acyclic. Also, we can conclude that if $q_a = 1$ then the scarcity condition is trivially satisfied for a Kesten cycle and if $q_a = q_b = 1$ then the scarcity condition is trivially satisfied for an Ergin cycle. For further explanations and examples to illustrate these definitions see Ergin ([13]) and Kesten ([21]). From this point, the strict ranking over three agents in the cycle is called the *long chain* and the strict ranking that is over two agents is called the *short chain*.

Proposition 4. *If a priority profile is Kesten acyclic, then it is also Ergin acyclic.*

This implication was first proved by Kesten (2006 [21]) by referring to multiple other theorems. We provide here the first direct proof of this result. Note that the converse implication does not hold: that is, while we show that having an Ergin cycle implies having a Kesten cycle, having a Kesten cycle does not imply that there is also an Ergin cycle.

Proof. We show that whenever there is an Ergin cycle, it is impossible to have Kesten acyclicity. We know that if $q_a = q_b = 1$ in order to check Ergin and Kesten cycles, only the loop condition needs to be considered. It is easy to see that the loop conditions are the same in this case. Therefore, without loss of generality, an Ergin/Kesten loop can be written as: $i \succ_a l \succ_a j$ and [$j \succ_b l \succ_b i$ or $j \succ_b i \succ_b l$]. In each of the preceding cases we assume that there is an Ergin cycle and suppose that the profile satisfies Kesten acycliciy. We will show that this leads to an impossibility in both cases. In both cases if $q_a = 1$ then there is a Kesten cycle, so in order to avoid a contradiction we can assume $q_a > 1$. We consider two possible cases.

1. $j \succ_b l \succ_b i$: The only possible case in which we have Kesten aciclicity is when for any feasible N_a^e we choose, there is at least one member say k, which is not in N_a^k. This is possible only if the profiles are $i \succ_a k \succ_a l \succ_a j$ and $k \succ_b j \succ_b l \succ_b i$. In this case we have two other Kesten loops which are $k \succ_b j \succ_b i$ with $i \succ_a k, j$ and $j \succ_b l \succ_b i$ with $i \succ_a l, j$. Therefore we have a Kesten loop but, for $q_b > 1$, this is not a Kesten cycle yet. Since we have not checked yet the scarcity condition. Now we should be able to find a proper N_b^e . Since N_a^e and N_b^e are disjoint sets, which are above i in $b's$ rank. We will call one of these agents h. If this agent is above j then we have a Kesten cycle: $j \succ_b l \succ_b i$ with $i \succ_a l, j$. If we want to avoid having a Kesten cycle, at least one of the agents, say h, must be below j in $b's$ priority: $h \succ_a i \succ_a k \succ_a l \succ_a j$ with $k \succ_b j \succ_b h \succ_b l \succ_b i$ or$k \succ_b j \succ_b l \succ_b h \succ_b i$. But even in this case there is a Kesten cycle, since $i \succ_a l \succ_a j$ and $j \succ_b l, i$. Therefore, a Kesten cycle will exist in any possible scenario and we can choose h to be a member of N_a^k. Therefore, the Kesten scarcity condition is satisfied in every possible

scenario.

2. $j \succ_b i \succ_b l$: In order to have Kesten aciclicity we need to show that for any choice of N_a^e there is going to be at least one member that is not in N_a^k. This happens when there is an agent, such as k, such that $i \succ_a k \succ_a l \succ_a j$ and $k \succ_b j \succ_b i \succ_b l$. But then there is going to be a Kesten Loop since $k \succ_a j \succ_a i$ and $i \succ_b k, j$. This Kesten Loop may be a Kesten cycle: in order to have an Ergin cycle we should be able to have some set N_b^e that satisfies the Scarcity Condition.That means that if $q_b > 1$ then there should be $q_b - 1$ agents above i (excluding k). If one of these agents, say m, is excluded from N_b^k then we have Kesten acyclicity. There are three possible rankings for m :

   (a) If $h \succ_b k \succ_b j \succ_b i \succ_b l$ then $k \succ_b j \succ_b i$ and $i \succ_a k, j$. Therefore, we have a Kesten cycle.

   (b) If $k \succ_b j \succ_b i$ and $i \succ_a k, j$ and $k \succ_b j \succ_b i$ and $i \succ_a k, j$. This case depends on the position of h in a's priority. If $h \succ_a i \succ_a k \succ_a l \succ_a j$ then there is another Kesten cycle which is $i \succ_a l \succ_a j$ and $j \succ_a i, l$. And if h is below i in $a's$ priority, then the cycle is going to be $k \succ_b j \succ_b i$ and $i \succ_a k, j$.

   (c) If $k \succ_b j \succ_b m \succ_b i \succ_b l$ then if in a's priority h is above l then $i \succ_a l \succ_a j$ and $j \succ_b i, l$ is a Kesten cycle. And if it is below l, then $j \succ_b h \succ_b l$ and $l \succ_a h, j$ is a Kesten cycle.

Therefore, it is not possible to have Kesten acyclicity. Hence the statement is proved. $\square$

3.5 Fairness Comparisons in Many-to-One Matching

As shown by Proposition4, an Ergin cycle is stronger than a Kesten cycle. We can show that with a Kesten cycle, which is the weaker version of the two cycles, we cannot find any profile in which one of the mechanisms has fewer justified envy instances in all possible preference profiles than any other mechanism.

Theorem 8. *Suppose that $\succ$ has a Kesten cycle. Let $|A| \geq 4$ and $|O| \geq 3$, and a fix an arbitrary quota vector q. Let f be a Pareto-efficient and strategy-proof rule. Then there exists another strategy-proof and Pareto-efficient rule g and a preference profile R such that f has at least one justified envy instance at R and g does not.*

This is a generalization of Theorem 1 which we proved in Chapter 2. Theorem 1 was established in a one-to-one matching model; now we extend it to a many-to-one model with multiple quotas.

Proof. First we prove the statement for four agents and then we generalize the proof to more than four agents.

Consider the loop condition in the priority profile. Each loop condition has two sets of strict priorities for each of the two objects. We will call the object with the larger set of strict priorities object a, and we say that object a has a long chain of strict priorities, i .e. $i \succ_a l \succ_a j$. likewise, the object with a short chain will be called b, i.e. $j \succ_b i$].If $q_a = q_b = 1$ then

2	1
a	b
i	$\cancel{k}$
k	j
l	i
j	l

(a) Case 1: k can be anywhere but first in $b's$ priority

2	1
a	b
i	$\cancel{k}$
k	j
l	l
j	i

(b) Case 2: k can be anywhere but first in $b's$ priority

2	1
a	b
k	...
i	j
l	i
j	l

(c) Case 3: k can be anywhere in $b's$ priority

2	1
a	b
k	...
i	j
l	l
j	i

(d) Case 4: k can be anywhere in $b's$ priority

Table 3: Four possible priority tables when $q_a = 2$ and $q_b = 1$

the result follows from Theorem 1 in Chapter 2. Thus, we assume either $[q_A = 2$ and $q_b = 1]$ or $[q_1 = a$ and $q_b = 2]$.

Also, we assume that the object with the long chain cannot have a quota of greater than two, since in this case the scarcity condition will not hold.

Part 1: First we assume that $q_a = 2$ and $q_b = 1$. In this case we have four possible priority tables (see Table 3).

Now we consider the case in which all agents prefer a to b. It g is a TTC based on this profile then the matching is going to be $(a, \varnothing, b, a)$. Suppose $f(R) \neq (a, \varnothing, b, a)$. For mechanism f all the higher priority agents for object a will be matched until the quota for this object becomes is filled. If not, then unfilled spot in a and the agent who has been matched to object b will form justified envy with this spot. The other agent is going to be matched with b, otherwise b can form justified envy with its highest priority

1	2
a	b
i	j
l	i
j	l

(a) Case 1 : The case that needs to be evaluated on the second part of the proof

1	2
a	b
i	j
l	l
i	i

(b) Case 2 : we have large chain on b as well, no need to prove it again

Table 4: Two possible priority tables when $q_a = 1$ and $q_b = 2$

agent among the remaining agents. Therefore, $f(R) = g(R) = (a, \varnothing, b, a)$. Now let $R'_i = (b, a)$ and $R' = (R'_i, R_{-i})$. Since the mechanism is strategy-proof, i is matched to either a or b. And since it is Pareto-efficient, it is matched to b. Therefore we have three different scenarios. These scenarios can be written as: $f_{(i,l,j,k)}(R) = (b, \varnothing, a, a)$, $f_{(i,l,j,k)}(R) = (b, a, \varnothing, a)$, or $f_{(i,l,j,k)}(R) = (b, a, a, \varnothing)$. In the first case, $f_{(i,l,j,k)}(R) = (b, \varnothing, a, a)$ means that j and k will receive a and i will receive b, and thus l will not be matched to any object.

Case 1: $f_{(i,l,j,k)}(R') = (b, \varnothing, a, a)$

This means that agent l will not be matched. Here (l, a, j) is a justified envy instance. Suppose we change $j's$ preference to (a, c, b). This will not change $j's$ allocation since the mechanism is strategy-proof and j is the least priority agent for a. Hence there is always a justified envy instance. Now if we choose a Serial Dictatorship, which is Pareto-efficient and strategy-proof with permutation (i, l, k, j) then the final matching is going to be $g_1(R'_1) = (b, a, c, a)$, which is fair.

Case 2: $f_{(i,l,j,k)}(R') = (b, a, \varnothing, a)$

Here (j, b, i) will form a justified envy instance. Like in the previous case, we put c between b and a in $i's$ preference, which will not change the final allocation of i. Therefore the same justified envy exists. But if we run Serial Dictatorship with (k, l, j, i) then this mechanism is fair and $g_2(R_2') = (c, a, b, a)$.

Case 3: $f_{(i,j,l,k)}(R) = (b, a, a, \varnothing)$

Similar to previous cases, (k, a, j) is a justified-envy instance. Now if we put c in the middle of j's preference, this will not change j's allocation. However, if we take any Serial Dictatorship mechanism which is based on k, i, l, j this mechanism is going to produce a fair matching at this profile.

Part 2: Now we assume that $q_a = 1$ and $q_b = 2$. In this case we have two possible profiles. If $i \succ_a l \succ_a j$ then either $j \succ_b i \succ_b l$ or $j \succ_b l \succ_b i$. If $j \succ_b l \succ_b i$ then the case is the same as before so we can assume $j \succ_b i \succ_b l$.

First we assume that all agents prefer b to a. With the same reasoning that we used before, we can prove that our mechanism should have the same outcome as the TTC mechanism. Then we consider two different conditions. $i \succ_a k$ and $k \succ_a i$. First we assume that $i \succ_a k$ and the other case will be proved in a similar way. If $i \succ_a k$ then i is definitely matched. Now we flip $i's$ preference, so i prefers a to b. Then the mechanism is going to have three possible outcomes. $(a, \varnothing, b, b), (a, b, \varnothing, b)$ or $(a, b, b, \varnothing)$.

Case 1: $f_{(i,l,j,k)}(R) = (a, b, b, \varnothing)$

If $k \succ_b l$ then (k, b, l) is a justified envy instance. Now if we put c in the middle of $l's$ preference,$R_l = (b, c, a)$, this will not change $l's$ allocation.

Then if k is matched to a then (i, a, k) is a justified envy instance and if k is matched to b then j can not be matched to b and then j and b can form a justified envy instance. Now any Serial Dictatorship with $j, i, k, l)$ produces a fair outcome at this profile. If $l \succ_b k$ then b is going to be matched to j and the mechanism can be treated as having a quota of $(1, 1)$, which has been proved before.

Case 2: $f_{(i,l,j,k)}(R) = (a, b, \varnothing, b)$

In this case (j, b, l) is a justified envy instance. Now if we put c in the middle of $l's$ and $k's$ preference,$R_{l,k} = (b, c, a)$, this will not change the allocation when applying TTC. But if we apply the Serial Dictatorship based on (j, i, k, l) the final matching is going to be $g'_{(i,l,j,k)}(R') = (b, \varnothing, b, a)$ a and the mechanism is going produce a fair outcome on this profile.

Case 3: $f_{(i,l,j,k)}(R) = (a, \varnothing, b, b)$

If $k \succ_b l$ then this case is going to be fair. In this case j is going to be matched with b and we can assume that we have quota of $(1, 1)$ which has been proved before. If $l \succ_b k$ then (l, b, k) is a justified envy instance. Therefore we can put c in the middle of k and $j's$ preference ,$R_{j,k} = (b, c, a)$, which will not change the final allocation of TTC. But if we run the Serial Dictatorship based on (j, i, l, k), the rule is going to assign a fair matching to this profile.

If $k \succ_a i$: We can use the same reasoning as for the previous case to prove the result.

Part 3: Different quota distributions

If we have more than four agents, then the quota needs to be increased proportionally in order to have cycles. But in any distribution at least one of the intermediate agents is going to be matched to the object with the long chain. Therefore, we can assume that we have the same problem but with the quota reduced by one and one fewer agents. This process can be iterated until we reach the distribution of $(2, 1)$ or $(1, 2)$, which have been proved before. $\square$

3.6 Simulations

We know that if there is a cycle in the priority profile, then it is not possible to have a fair and Pareto-efficient matching at all preference profiles. Therefore, following the definition of Kesten cycles, we can count the number of cycles in each priority profile. We want to study the effect of having more cycles. Is there a difference between the priority profiles that have more cycles and the priority profiles that have less? As we see from an example, there are unexpected profiles in which the priority profile which has more cycles is fair, and the one with fewer cycles has justified envy instances compared to the former. But, as we show by simulation, there is a reliable trend that shows that as the number of cycles increases, the chance of having more justified envy instances for the TTC allocation increases and the number of preference profiles that will cause justified envy instances will increase as well. Table 5 shows that we cannot theoretically conclude that a mechanism with more cycles always has fewer justified envy instances. In case 1 the priority profile has three different cycles: the first cycle is $k \succ_b l \succ_b j$ with $j \succ_a l, k$, the second cycle is $l \succ_b i \succ_b j$ with $j \succ_a l, i$, and the last cycle is $k \succ_b i \succ_b j$ with $j \succ_a i, k$ and the scarcity condition is satisfied. If we consider this priority profile with the given preference profile (see Table 5), the mechanism is going to be fair: m and i are matched to a and k is matched to b.

If we consider the priority profile in case 2, we have only two cycles: $l \succ_a j \succ_a m$ with $m \succ_b l, j$ which satisfies the scarcity condition since $N_a^k = \{i\}$ and $m \succ_b j \succ_b l$ with $l \succ_a j, m$. This priority table with the same preference profile has the following allocation: i and m are matched to a and l is matched to b. Here we have one justified envy instance: (j, b, l). Therefore, we cannot analytically prove that more cycles will lead to more justified envy instances, as it is not true. However, we will show by using simulation that in most of the cases more cycles do lead to more justified envy instances.

Figure 1 shows the results of simulations for 100 preference profiles when the quota vector for two objects is $(2, 1)$. There are five different agents, and agents can choose to stay unmatched at any stage of the mechanism. The simulation is for 100 different priority profiles. The priority profiles are random permutations of agents. In each priority profile we count the number of cycles. Then we generate 100 random preference profiles, which is a permutation over all objects plus the case that agent will prefer to stay unmatched, with a constraint that in the simulation each agent prefers at least one object. and we count the number of preference profiles over which there is no justified envy instance. Then we do the same simulation for the aggregate number of justified envy instances for each of these priority tables over 100 random preference profiles.

As can be seen, as the number of cycles increases, the number of profiles in which the TTC mechanism has fewer justified envy instances decreases. However, recall that this is different from the definition of JE-domination, since we are considering only a limited number of preference profiles, and as we have shown in our preliminary example, although this trend holds, it is not true for all preference profiles.

However, there is a general trend that shows that whenever there are more cycles in the priority table the number of preference profiles under which the TTC mechanism is fair decreases. Moreover, the aggregate number of JE-instances increases as the number of cycles in each priority profile increases.

In Figures 1 and 2 there are 100 random priority profiles and 100 random preference profiles. For each priority table we count the number of cycles and we show it on the horizontal axis. Then for each priority table we run the TTC mechanism for 100 random preference profiles. In Figure 1 the horizontal axis indicates the number of cycles in each of these 100 priority profiles and the vertical axis shows the number profiles for which TTC has justified envy instance. As a general pattern in the simulation, as the number of cycles increases, there are more profiles in which the mechanism is not fair. In Figure 2 the number of justified envy instances is graphed with respect to the number of cycles. Again, there is a positive correlation between the two variables. Tables 6 and 7 show that these linear regressions are reliable. Therefore, we can conclude that there is a negative relation between the number of Kesten cycles and preference profiles for which the

2	1
a	b
m	k
j	m
k	l
l	i
i	j

(a) Case 1: Priority table with three Kesten cycles

2	1
a	b
i	i
l	m
j	j
m	l
k	k

(b) Case 2: Priority table with two Kesten cycles

i	j	k	l	m
a	b	b	b	a
$\varnothing$	$\varnothing$	a	$\varnothing$	b

(c) Preference profile in which we can see this irregular behavior

Table 5: An example that shows that simulation is needed to see the relation between the number of cycles and JE-instances

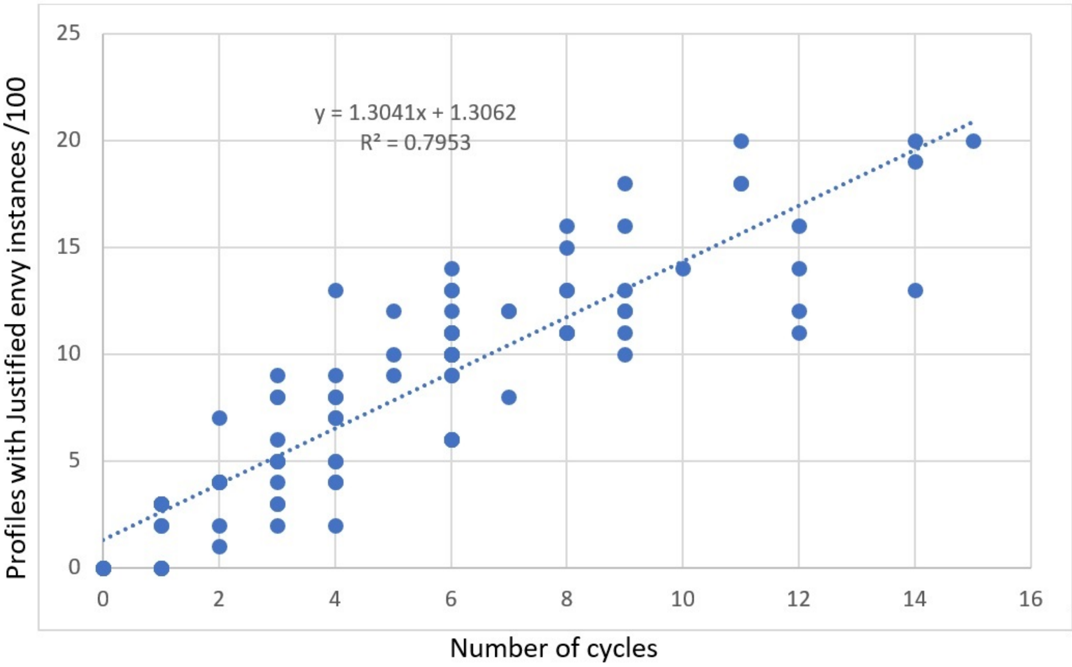


Figure 1: Preference profiles for which the TTC is fair vs. the number of Kesten cycles

TTC mechanism is fair, and a positive relation between the number of justified envy instances in the TTC and the number of cycles.

profiles	Coef.	Std. Err.	t	$p > \|t\|$	[95%	$Confidence Int.]$
Slope	1.304077	0.0668294	-19.51	0.000	-1.436697	-1.171456
Intercept	1.3062	0.432485	228.20	0.000	97.83552	99.55202

Table 6: Confidence interval and reliability test for Figure 1

profiles	Coef.	Std. Err.	t	$p > \|t\|$	[95%	$Confidence Int.]$
Slope	2.024872	0.0935502	21.64	0.000	1.839225	2.210519
Intercept	0.5071833	0.6054082	0.84	0.404	-0.6942296	1.708596

Table 7: Confidence interval and reliability test for Figure 2

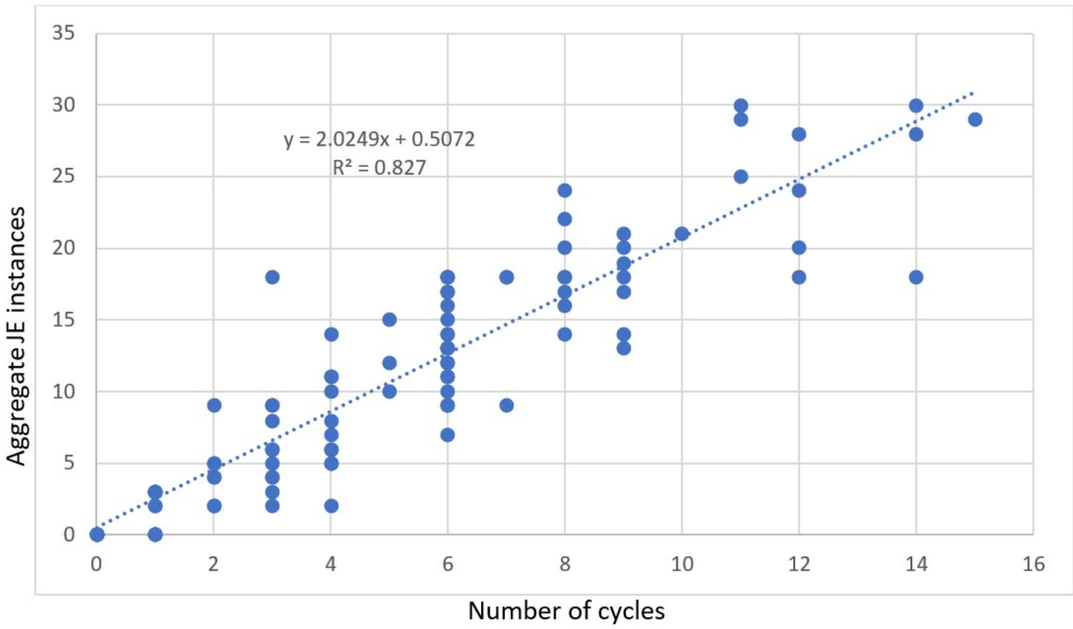


Figure 2: Aggregate JE-instances in the TTC vs. the number of Kesten cycles

3.7 Conclusion

When there are multiple quotas on the object side, in the acyclicity condition for priority profiles scarcity needs to be considered as well. In many-to-one models there is a difference between Ergin and Kesten acyclicity. We show that a Kesten cycle is a weaker version of an Ergin cycle. But even with this weaker condition there is no strategy-proof and Pareto-efficient mechanism which has fewer preference profiles that are fair, when considering each profile side-by-side, than any other strategy-proof and Pareto-efficient mechanism.

We have also run simulations to show that there is a negative relation between cycles and preference profiles for which the TTC gives a fair outcome. In this linear regression the $R\!-\!squared$ is 0.7953, and the number of observations is 100. Also, as the number of cycles increases, the aggregate number of JE instances increases as well.

Chapter 4

Competing School Choice Mechanisms

4.1 Introduction

High quality public schools are scarce, and just like allocating any scarce valuable resource, we need economists to assign students to schools in a proper way. This need fostered a new strand in economics, known as market design. Historically, students were enrolled at schools based on having siblings in the school or living in the neighborhood of the school. This decentralized approach has several disadvantages, including segregation of students along socioeconomic lines and leaving room for unfair or inefficient rationing of school seats which may give incentives to students to misreport their true preferences.

In market design different schools are endowed with typically diverse priorities over prospective students, which may be based on the applicants' skills, abilities and/or compatibility with the school, or simply reflect priorities mandated by school boards, such as walk zone priority and priority due to having an older sibling at the school in the case of Boston schools. In turn, students have preferences over schools based on their understanding of curricula and extra curricula, the rate of graduation, the rate of future admission to universities, students' success in the job market and the distance from school, among others. In order to match heterogeneous students to schools in an effective way, students submit their preferences to a central clearinghouse. Several properties should be satisfied to have a desirable matching between students and schools. The most prominent properties are fewer appeals about the unfairness of the allocation, efficient allocation of schools with different appeal to students with various interests and talents, and minimum gaming and behind-the-closed-door strategies. Fairness and efficiency are the main normative criteria, while minimizing strategic behavior is also important.

A desideratum for parents and schools is receiving a fair matching. In the context of school choice, fairness, which is the lack of justified envy, is more important than stability (the property that is formally identical to fairness but rules out blocking pairs instead of justified envy instances), since the supply side of school seats is under public control and schools may not have much incentive to circumvent the market, while students or their parents can justify their envy based on the priorities set by schools, and may appeal or sue in case of a school priority violation. From their viewpoint, if a student prefers a school in which she has higher priority than a student who got matched to that school, then her envy is justifiable and the mechanism is not fair.

From the designer's viewpoint, the matching should be Pareto-efficient in the sense that if we can improve the assignment of some students without harming the assignment of others then the matching does not fulfill this property. Another important property is truth-telling or strategy-proofness. If students realize that by misreporting their preferences they can be better off, then they may game the system and gain unfair advantage over others, and it will also be impossible for the market designer to match students and schools according to the normative criteria, if the matching is based on false preferences. This point of view is applicable not only to schools and students, but also to every matching problem in which we have multiple copies of the objects on one side of the market.[1]

For the designer it would be best to find a mechanism that meets all these criteria, but it is well-known that this is not possible. In general, several competing matching mechanisms have been proposed in an attempt to make markets more efficient, fair and strategy-proof. But it is not possible to have all desired properties, since none of the mechanisms can fulfill all criteria. Roth (1982) [32] showed that we cannot have fairness and efficiency simultaneously for arbitrary school priorities. Kesten (2010) [22] proved that we cannot have a strategy-proof and Pareto-efficient mechanism which chooses the fair allocation whenever a fair Pareto-efficient allocation exists. Ergin (2002)[13] showed that in order to have an efficient and stable

[1]Here we consider misreporting of the preferences as the only way of manipulating the outcome. A student may also mislead schools in other ways in order to make them rank her higher than her real competence would allow.

mechanism the priority structure of schools needs to be *acyclic*. Kesten (2006)[21] provided another definition for *acyclicity* which enables having both stability and Pareto-efficiency simultaneously. Heo (2017) provides a maximal preference domain in which stability and efficiency are possible to achieve simultaneously. Priority structures under which the TTC and the Boston mechanism always lead to an efficient and fair outcome were given by Kesten (2006)[21] and Kumano (2013)[24], respectively. Abdulkadiroğlu et al. (2017 [3]) proved that none of the competing mechanisms that we compare in this paper are justified envy minimal inclusion-wise, with respect to blocking pairs. There is also a growing literature on weakening stability notions that are compatible with efficiency.[2]

Among the properties that we have discussed so far, Pareto-efficiency and strategy-proofness are in some sense the most important: having a blocking pair may not always lead to winning the lawsuit, since if the matching is not Pareto-efficient, some students might be worse off (Morril, 2015-a) [26]. Therefore, this contention will not be resolved. Strategy-proofness in addition is very important since efficiency and fairness are based on reported preferences, and therefore if reported preferences are not true, we cannot be sure that normative properties can be satisfied.

There are some previous papers that compare different mechanisms. Morril (2013, [26]) compares two mechanisms, CT and PTC and their combination[3], with the TTC. We will introduce the CT (Clinch and Trade) rule

[2]See, for example, Kesten, 2004 [20]; Cantala and Pápai, 2014 [8]; Alcade and Romero-Medina, 2015 [4]; Dur et al., 2015 [11]; Morrill, 2015 [27]; Kloosterman and Troyan, 2016 [23]; and Ehlers and Morrill, 2017 [12]

[3]In the PTC (Prioritized Trading Cycles) rule each school points to its highest average rank student among the top q_a acceptable students.

in detail later, which allows students to "clinch" a school if they have high enough priority for the school (to be defined precisely), even if the student does not have the current top priority, as required by the TTC. Morrill shows that a combination of CT and PTC give us the best result. He works with different environments, but always with an equal distribution of seats. Hakimov and Kesten (2014 [15]) provides an experimental comparison between ETTC and TTC. ETTC was also introduced as a competing rule to TTC. It goes a step further than CT and allows students with high enough priorities not only to clinch schools but also to trade them, another feature which is not allowed by the TTC, since the TTC assigns each copy of an object as "endowment" to the top priority agent only, and thus in the TTC rule only the top priority agent for each object can clinch or trade the object in each step of the iterative procedure.

In this paper we compare ETTC with CT, as well as ETTC and a variation of CT called FCT, by testing these matching rules in different scenarios and environments. We also consider the effect of having different distributions of seats and their implication on the outcomes of the mechanisms. This is the first paper, to our knowledge, which tests different quota distributions in a school choice model.

In our simulations the preferences are not completely independent since in our formulation we allow for common information in the market in the form of a common ranking, which influences each student's rankings of schools. This results in a positive correlation among the preferences of students, which is not only a realistic feature of school choice preferences (and of many other applications) but also causes the same sample size to be a better measure of the population than in most previous studies.

We simulate four competing mechanisms which are strategy-proof and Pareto-efficient but not fair, and check which one has less *aggregate justified envy* (henceforth, AJE) compared to the other rival mechanisms. Aggregate justified envy numbers show the number of justified envy instances in total across all sampled preference profiles. Although we can show that none of the alternative mechanisms are superior at every single possible profile, if the AJE for a specific mechanism is significantly lower, then the probability that participants have justified envy and thus perceive that the mechanism is not fair is lower. We count AJE by simulation across a large number of randomly generated preference profiles in our setup.

In the first part of the paper, we show by using simulation that among the mentioned mechanisms, ETTC performs relatively better in terms of having fewer AJE. In addition, we compare the rate of success of different mechanisms in terms of having lower AJE when we have different distribution of quotas and different levels of common information among students. In the second part of the paper we consider the quota distribution of the schools. We show that if higher ranked schools have higher quotas, AJE decreases for each studied mechanism and in addition more students are matched with the schools that are ranked higher by them. Therefore, if the school seat distribution in school choice design can be considered an endogenous parameter, then our findings suggest to choose an unequal distribution of quotas in order for the school choice mechanisms to better fulfill our requirements.

4.2 Model

We have a finite number of students and schools, given by $A = \{a_1, a_2, ..., a_n\}$ and $O = \{o_1, o_2, ..., o_m\}$, respectively. Each school o_j has a specific quota of students $q_{o_j} \geq 1$. Students have strict preferences over schools and schools have strict priorities over students.

For each $a \in A$ we will write $o_n \mathcal{P}_a o_m$ if student a strictly prefers o_n to o_m. With similar notation we will write $a_n \mathcal{P}_o a_m$ if school o has a higher priority for a_n than for a_m. A preference and priority profile in a matching market is written as $\mathcal{P}$, and the set of profiles is denoted by $\mathbb{P}$. An assignment μ is a function from students to schools: $\mu : A \to O$ such that for each student a_i and each school o_j that are matched to each other: $\mu(a_i) = o_j$ and $a_i \in \mu^{-1}(o_j)$, where $|\mu^{-1}(o_j)| \leq q_{o_j}$. The set of all possible matchings is denoted by $\mathbb{M}$. A mechanism ν is a function from all possible profiles to the set of matchings: ν: $\mathbb{P} \to \mathbb{M}$.

4.2.1 Normative Criteria

A mechanism eliminates *justified envy* if there does not exist any pair of student a and school o at any profile $\mathcal{P}$ such that $o\mathcal{P}_a\mu(a)$ and $a\mathcal{P}_o b$ where $b \in \mu^{-1}(o)$ and $\nu\mathcal{P}) = \mu$. If such a pair exists then the student's envy of the school is justified, but if the school does not have higher priority for such a student then, although the student has envy, her envy is not considered justified.

A matching μ is *Pareto-efficient* at profile $\mathcal{P}$, if there is no matching μ' such that at least for one student a we have $\mu'(a)\mathcal{P}_a\mu(a)$ without having $\mu(a')\mathcal{P}_{a'}\mu'(a')$ for any other student a'. A mechanism is Pareto-efficient if it assigns a Pareto-efficient matching to each profile.

A mechanism is *strategy-proof* if when the true preference of student i is $\mathcal{P}_i$ there is no preference profile $\mathcal{P}'_i$ such that $\nu_i(\mathcal{P}_{-i}, \mathcal{P}'_i)\mathcal{P}_i\nu_i(\mathcal{P}_{-i}, \mathcal{P}_i)$.

4.2.2 Correlated Rankings

In order to make our sample more realistic, we assume that the students' rankings are affected by a common ranking, which can be interpreted as having some common information over schools. There are some previous papers on correlations in the environment and their effect on the properties of the mechanisms (see for example, Abdulkadiroğlu et al.(2011 [1]) and Karpov (2017 [19]). An additional technical advantage of assuming such common information over schools is that it allows us to have a proper sample size. Our possible sample of preference profiles grows as the number participants in the market grows, and due to computational limitations considering all possible profiles is not feasible. We will write the aggregate ranking in this way to reduce the population size and to make our sample more accurate.[4]

[4] Some conclusions in the second part of the paper depend on this presentation of rankings.

The common ranking is a fixed ranking of schools. It could be interpreted as some common information available to all students. By contrast, the individual ranking is a ranking which is generated randomly for each student in our simulation and reflects each student's individual taste.

For each of these rankings we give the score of one to the least preferred school and we increase the score by one unit as we go from last to most preferred. In order to find the aggregated ranking we use different weight distributions over the common ranking and the individual rankings. This weight is given by $\alpha \in [0, 1]$ in our formula as follows: $(\alpha \times \textit{common ranking} + (1 - \alpha) \times \textit{individual ranking})$. The higher is α in this formula, the more the rankings are correlated. We add the score of each school in different rankings while taking into account the weights. The highest-ranked school is the one with the highest score. If we have any ties, we break the ties in favor of the common ranking. Note that for the schools' side we do not use this correlation since the priorities are imposed by law.

4.2.3 Gini Coefficient

There are many possible distributions of seats among the schools. One possible distribution is where the seats are equally distributed. On the other hand, seats could be distributed in favor of one or more school more than others. The two ends of the distributions in the spectrum of possible distributions are equal seats for all schools at one end, while at the other end one seat is given to each school except the most favored one and the remaining seats go to the favorite school according to the common ranking. In order to be able to compare different distributions of seats we use a concentration index. One very common index is the Gini coefficient which varies between 0 and 1. The closer it is to zero the closer it is to equal distribution of seats. This measure is usually used as a measure of inequality and concentration of income or wealth. If one school has all the seats and the rest have zero then the Gini coefficient is equal to one. In our simulation the minimum number of possible seats is one seat for each school.[5] Therefore, in this paper the maximum Gini coefficient cannot be one.

Suppose that there are five schools and twenty seats in total. If we normalize the number of schools and seats to one, then each school is 0.2 of total schools and each seat is 0.05 of total seats. Then we sort the schools from lowest to the highest quota on horizontal axis and the seats on vertical axis. We have a kinked line which is called the Lorentz curve. (In thick markets this kinked line looks more like a curve.) The area under this line is shown by B (as shown in Figure 1). Moreover, there is a straight line which shows the scenario in which all schools have an equal number of seats. The area between straight line and the Lorentz curve is A, as indicated in Figure 1. The Gini coefficient is $\frac{A}{A+B}$.

As a case in point if four out of five schools have a quota of one and the last school has a quota of 16, the Gini coefficient is going to be 0.6.(As shown in Figure 2.)

[5]Otherwise we can ignore the school and run the matching procedure without this school.

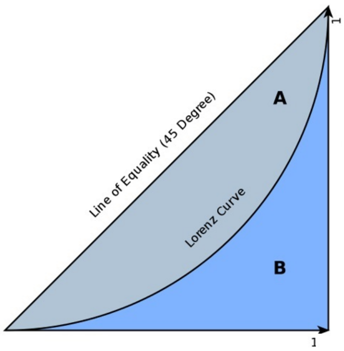


Figure 3: Gini coefficient

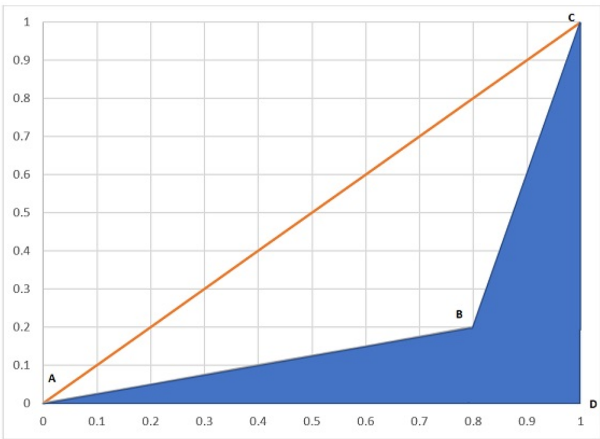


Figure 4: Gini coefficient when the quota distribution is (1,1,1,1,16)

4.3 Competing Mechanisms

4.3.1 TTC Mechanism

Abdulkadiroğlu and Sönmez (2003) adopted a generalized version of Gale's top trading cycle mechanism for school choice. The original TTC mechanism was first used for allocating agents to objects in a model with a one-to-one endowment of objects by Shapley and Scarf (1974). The procedure is as follows:

- Each student points to his highest ranked school.
- Each school points to its highest priority student.
- Since we have a finite number of students and schools, there is at least one cycle.
- We allocate schools to students according to the cycles.This allocation is final and the matched pairs are removed from the market.
- We iterate this procedure until there is no student or school remaining in the market.

Proposition 5 (Abdulkadiroğlu and Sönmez, 2003 [2]). *The TTC mechanism for school choice is Pareto-efficient and strategy-proof.*

However, in many cases this mechanism is not fair. Hence, there are some attempts in the literature to change the mechanism and make it fairer, that is, reduce justified envy. One downside of TTC is the fact that schools involve *only* their highest priority student in the cycles in each round, despite having multiple seats. This approach makes the market thin and only students with the highest priority participate in the formation of each cycle. If we could make the market thicker we would give a chance to students who do not have the highest priority to be assigned to better schools. Specifically, we expect that this leads to fewer justified envy instances. For this reason, desirable alternative mechanisms have been proposed that may reduce justified envy instances, which we introduce next.

4.3.2 FC&T and C&T Mechanisms

Morrill (2015-b) proposed two mechanisms, FC&T and C&T. In FC&T, in the first step if school o with a capacity of q_o has one of the q_o highest priorities for a student and the student has the highest preference for the school then the student can clinch the school. In the second step we run the regular TTC mechanism.

In C&T mechanism clinching is allowed in every step and not just initially. If a school has been deleted, then for all the students that are pointing to the deleted school(s), we run the clinching mechanism again. We repeat the regular TTC afterwards until there is no student or school that is left unmatched.

Proposition 6 (Morrill, 2015-b)**.** *The FC&T and C&T mechanism are both Pareto-efficient and strategy-proof.*

4.3.3 ETTC Mechanism

The equitable top trading cycle mechanism (ETTC) was introduced by Hakimov and Kesten (2014).

- Each school o with a quota of q_o forms q_o pairs with its top q_o priority students.
- If a student has her favorite school in one of her pairs, she points to her own pair.
- In each pair, the student points to the pair that contains her highest ranked school.
- If there are several pairs that fulfill this property, the student points to the pair that has the student who has highest priority for the school in the pointing pair.
- Since there is a finite number of pairs, we have at least one cycle.
- Assign students to the schools they are pointing to in each cycle and remove assigned students and schools.
- After each cycle, since some students can form different pairs and after the trade in one pair students are not available anymore, we need to delete all other pairs that this student is in.
- School slots that remain unassigned after students are removed from the market are inherited under certain conditions, namely, there should be no other student who is paired with this school in that

stage. Once all such students are assigned in the market, the remaining seats will be inherited by other students according to the priority of schools. This restriction on inheritance guarantees that the mechanism is strategy-proof.

- We repeat these steps until no more students can be matched.

Proposition 7 (Hakimov and Kesten, 2014 [15])**.** *ETTC is Pareto-efficient and strategy-proof.*

Proposition 8 (Hakimov and Kesten, 2014 [15])**.** *If there are only two schools and student a is in the top q_o priorities of school o, then student a does not have justified envy under ETTC.*

Proposition 9 (Hakimov and Kesten, 2014 [15])**.** *Suppose thee are only two schools. Then if an assignment is not fair at some profile under ETTC, then the assignment is not fair under TTC at the same profile.*

4.4 No Superior Mechanism

Although it might appear that the modified mechanisms improve upon the TTC mechanism at each profile, at some profiles they lead to more justified envy, when there are multiple seats. Note that if the number of seats is one for each school, all mentioned rival mechanisms are the same. In order to demonstrate that TTC is not worse for all preference profiles and, in fact, may be better than other mechanisms, we provide examples where TTC has fewer justified envy instances than C&T/FC&T and ETTC.

4.4.1 TTC vs. ETTC

Consider the profile in Table 8 with quota distribution (2,1,1). The final matchings for ETTC and TTC show that TTC does not have any justified envy, whereas ETTC has one justified envy instance with student 4 and school 1.

4.4.2 TTC vs. C&T/FC&T

Consider the profile in Table 9 with a quota of 2 for both schools. The final allocations under C&T/FC&T and TTC are shown in Table 9. It can be seen that C&T/FC&T has one justified envy instance with student 2 and school 1, whereas the TTC mechanism has no justified envy at this profile.

4.4.3 ETTC vs. C&T/FC&T

In some profiles ETTC has a relative advantage and in some others C&T/FC&T has the advantage. In Table 10, in the first set of profiles ETTC has a relative advantage over C&T/FC&T. In this mechanism C&T has one justified envy instance with school 3 and student 2, whereas ETTC does not have any.

In another example, with a slight change in priorities of the schools, C&T/FC&T has relative advantage over ETTC. In Table 11 the only justified envy instance is student 1 and school 2 for ETTC, and C&T/FC&T

2	1	1
o_1	o_2	o_3
a_1	a_1	a_2
a_2	a_3	a_3
a_4	a_4	a_1
a_3	a_2	a_4

(a) School Priorities

a_1	a_2	a_3	a_4
o_3	o_1	o_1	o_1
o_2	o_3	o_2	o_3
o_1	o_2	o_3	o_2

(b) Student Preferences

Schools	3	1	2	1
Students	1	2	3	4

(c) Matching under TTC

Schools	3	1	1	2
Students	1	2	3	4

(d) Matching under ETTC

Table 8: A profile at which ETTC has more justified envy than TTC

2	2
o_1	o_2
a_4	a_3
a_3	a_1
a_2	a_2
a_1	a_4

(a) School Priorities

a_1	a_2	a_3	a_4
o_1	o_1	o_1	o_2
o_2	o_2	o_2	o_1

(b) Student Preferences

School	1	2	1	2
Student	1	2	3	4

(c) Matching under TTC

School	2	1	1	2
Student	1	2	3	4

(d) Matching under C&T/FC&T

Table 9: A profile at which C&T/FC&T has more justify envy than TTC

2	2
o_1	o_2
a_4	a_2
a_3	a_1
a_2	a_3
a_1	a_4

(a) School Priorities

a_1	a_2	a_3	a_4
o_2	o_1	o_2	o_2
o_1	o_2	o_1	o_1

(b) Student Preferences

School	2	1	2	1
Student	1	2	3	4

(c) Matching under ETTC

School	2	1	1	2
Student	1	2	3	4

(d) Matching under C&T/FC&T

Table 10: A profile at which C&T/FC&T has more justified envy than ETTC

2	2
o_1	o_2
a_4	a_2
a_3	a_4
a_2	a_1
a_1	a_3

(a) School Priorities

a_1	a_2	a_3	a_4
o_2	o_1	o_2	o_2
o_1	o_2	o_1	o_1

(b) Student Preferences

School	1	1	2	2
Student	1	2	3	4

(c) Matching under ETTC

School	2	1	1	2
Student	1	2	3	4

(d) Matching under C&T/FC&T

Table 11: A profile at which ETTC has more justified envy than C&T/FC&T

does not have any justified envy.[6]

4.5 Simulations

We use simulation in order to compare these competing mechanisms which are not superior to each other in terms of justified envy when we compare them profile by profile. We do comparative simulations to see the aggregate difference between mechanisms when using different samples with different configurations of quotas and correlations.

We test the mechanisms for five schools and different numbers of students. Our simulations are for 10, 15 and 20 students. Each simulation was replicated 10,000 times. When the number of students are 10 and 15, we assume that the quotas of all schools are equal. When we have 20 students, we test 3 scenarios: equal quotas (E), one school only with a large quota and others having only one seat (X), and the last environment is an example for an unequal, but not extreme, quota configuration (U). Specifically we will check three different quota configurations of five schools as follows:

$$\text{E:}(4,4,4,4,4); \quad \text{X:}(16,1,1,1,1); \quad \text{U:}(7,7,2,2,2)$$

For the extreme (X) and unequal (U) environments we assume that schools that are ranked higher in the common ranking have more seats.

[6] To the contrary of our expectation we can also show that in some profiles the assignment under FC&T gives us fewer justified envy instances in comparison with C&T.

This assumption is justified in the appendix where we compare results with the opposite assumption regarding school seat distribution for unequal configurations of quotas.

4.5.1 TTC Simulations

Table 12 shows the results of the TTC simulations. The first row shows the number of students and the first column shows the different levels of correlation between the common ranking and individual rankings as indicated by α. There are 5 schools and we ran 10,000 simulations for each scenario. For 10, 15 and 20(E) we have equal quotas for each school and for 20(X) and 20(U) the quotas are different as indicated above. Henceforth, each correlation will be called a scenario, and each configuration of quotas will be referred to as an environment. Thus, Table 5 displays 5 scenarios corresponding to the 5 rows, and 5 environments corresponding to columns 2-6. As an example, for the correlation of 0.65 in the extreme environment (X), when we have twenty students and five schools, the AJE is 28,536, meaning that in 10,000 simulations this is the number of instances of justified envy altogether.

From the table we can see that if there are many schools with similar quotas the possibility of having justified envy is a lot higher than in the cases with unequal distributions of quotas. Also, the extreme distribution scenario shows much less AJE then the corresponding unequal distribution scenarios. This suggests that when schools have a homogeneous distribution of seats, the possibility of having justified envy increases.

α	10	15	20(E)	20(X)	20(U)
0.5	34679	78158	132347	34584	75578
0.55	35545	77258	127598	34264	73580
0.6	38228	82746	134936	29626	79471
0.65	35792	79714	131653	28536	78943
0.7	23478	54989	90525	17339	57773

Table 12: AJE in TTC

4.5.2 FC&T and C&T Simulations

We tested the same scenarios and environments for both FC&T and C&T. Here we can see that there is a reduction in the number of AJE compared to TTC (see Tables 13 and 14). The numbers in the parenthesis show the percentage reduction of AJE compared to TTC. As an example, for ten students and five schools, if the quota for each school is equal to two and the correlation is 0.5, the AJE for TTC is 34,679 and for FC&T this number is 34,239 which is a 1.23% reduction in AJE: $\frac{34679-34239}{34679} = 0.0127 = 1.27\%$

A comparison of tables 13 and 14 reveals that C&T performs better than FC&T.

4.5.3 ETTC Simulation

We test the same scenarios and environments for ETTC to check instances of justified envy (see Table 15).

As we can see, the number of AJE is less than in the TTC.

In all cases with an equal distribution of quotas ETTC has the smallest number of justified envy. On the other hand, with an unequal configuration of seats, as the Gini coefficient grows C&T and FC&T become more successful compared to the case with the equal distribution of seats, and as we get closer to the extreme quota distribution they become better than ETTC.

In Table 16 we test

$$H_0 : \text{justified envy instances}_{CT} = \text{justified envy instances}_{ETTC}$$

and

$$H_\alpha : \text{justified envy instances}_{CT} > \text{justified envy instances}_{ETTC,}$$

which justifies having more justified envy instances in CT. If these numbers are small, one possibility is that the difference in the average number of justified envy is not considerable, or the sample size might be small. But as we can see, since the t-test gives us large numbers, we can conclude that the sample size is large enough to deduce that the difference between justified envy instances is not biased by our sample size.

4.6 Efficiency

Next, we compare the efficiency of the mechanisms in the different configurations of quotas. We find the percentage of instances in which a student is matched with her first to fifth preferred choice of schools. We do this for

20 students and 5 schools in the three main environments that we have analyzed before, E, X and U. The three different scenarios that we are using are 0.5, 0.6 and 0.7. As can be seen in tables 17 and 18, as the correlation increases the market becomes more competitive, and therefore the level of efficiency decreases. As a case in point in Table 17(a), where the correlation is 0.7, the percentage of students who have been matched to their first choice in TTC is 25.85% and when the correlation is 0.5, 40.23% of students will get their top choice. Secondly, as the Gini coefficient increases, more students will be matched to their top choice. Consider a correlation of 0.7 and TTC scenario for 10 students and 5 schools, with each school having two seats in three different scenarios: equal distribution of seats (E), unequal distribution of seas (U) and all schools having only one seat except one (X). The percentages of students who are matched to their top choices are 25.85%, 45.15% and 77.6% respectively. In the very extreme case (X), since the most favored school has maximum quota, when the correlation increases most of the students will receive their top choices. In the simulation for extreme unequal distribution of seats, we can see that having a higher correlation will lead to better outcomes, since there are enough seats for the most favored school, and therefore, even though many students like the same schools, there is going to be enough room for all of them.

We can define a measure of efficiency which enables us to compare the relative advantage of mechanisms in matching students to schools that are ranked highly by them in different scenarios and environments. When we have 5 schools and 10,000 simulations, each student gets $5-r$ points, where r is the rank of the school in the student's preferences. In order to normalize we divide the total points that students get by 40,000. Since the maximum total points that a student can get is 40,000 , this normalizes the efficiency measure to be between zero and one. By calculating this measure for different Gini coefficients, we check if it is increasing as the concentration index increases. The closer this number is to one, the more efficient the market is. This can help us to compare the efficiency of the different markets. Table 20 compares the efficiency measures for TTC, ETTC and C&T in three different scenarios and three different environments.

We can conclude that the more seats are concentrated in one school, the fewer AJEs will occur and the more efficient the assignments become. Given this conclusion, we recommend a new policy to give more seats to schools that are ranked higher according to the common ranking. However, there are natural constraints that don't allow for an extreme application of this policy, conditions such as having schools with a certain minimum number of seats in each district. Therefore, as much as the constraints allow, we propose a more unequal and more concentrated seat configuration in favor of high-ranked schools instead of having an equal distribution of seats.

4.7 Discussion

One of the weaknesses that TTC has in school choice is that it only incorporates the first priority of the schools in forming cycles. C&T and FC&T modify this by allowing students who are not the first priority of the schools but are in one of the top q priority choices to be matched with these schools before the formation of a trading cycle (called clinching). This approach makes the market thicker and the mechanism becomes more successful in reducing AJE.

In ETTC, when a student is paired with a school which is her highest preference, she can also clinch that school. Moreover, in ETTC students in the top q priority choices are also allowed to trade their school seats, as opposed to TTC, C&T, and FC&T and, when a student wants to point to a school, in case she has more than one option, she will use her own schools' priorities in the matching. Due to this, there are fewer justified envy instances in the final allocation. This mechanism is most effective when there are only two schools, or when cycles have only two members. In these cases schools' priorities are considered more successfully than in other cases.

To conclude, we briefly summarize below our findings and the reasons behind them.

1. Compared to C&T, ETTC has lower AJE.

   Reason: As we have shown in the examples, there are many preference profiles where FC&T and C&T have fewer JE over ETTC and many preference profiles that are the other way around. In order to compare the mechanisms, we have to compare the frequency of these justified envy instances across preference profiles. Therefore, we present a statistical comparison of the mechanisms. We also checked that the sample size is large enough and the sample is reliable by running a t-test for mean comparison. (Table 16).

2. When there is a correlation between the common ranking and the

individual rankings, if the Gini coefficient is high, then students are matched with higher-ranked schools (Table 20).[7]

Reason: If there is a high correlation between common ranking and individual rankings, more students will rank schools that have high common ranking on the top of their rankings. If we increase the quota of these schools, lots of students will point to them and the chance for students to be matched with these schools will increase. Therefore, in general, students will get matched with a higher-ranked school.

3. If the Gini coefficient is higher, all mechanisms are more successful in having lower AJE.

   Reason: In profile-by-profile comparison, when the students are matched with high-ranked schools, there are going to be fewer schools above the schools that are matched with them, therefore, there is less chance for having JE instances. (see Tables 21-23). In each graph, the horizontal axis is the Gini coefficient and the vertical axis is AJE, and each point in the graph represents 10,000 simulations.)

   This suggests a policy that if there is a choice among different environments in a school district, one which has more unequal distribution of seats, in favor of most popular schools, should be preferred.

4. As the correlation between common and individual rankings increases, all alternative mechanisms to TTC are more successful in terms of having lower AJE than TTC. However, if the Gini coefficient increases, this effect may be smaller or even reversed.

[7]The efficiency measure that we use here is different from Pareto-efficiency. Here we are considering the average rank of a school that is matched to students based on students' preference rankings.

Reason: In C&T, if the correlation is high, there is more chance to have clinching with higher-ranked schools. Therefore, there is more chance that those schools will be clinched, hence we have fewer AJE. However, if the Gini coefficient is high as well, this will make most of the agents be matched with their favourite schools in TTC, and there is not much room for having less AJE in C&T.

In ETTC, when student preferences are correlated, students will point to the same schools. Therefore, the probability of forming small cycles increases. This includes cycles with two pairs in which ETTC is more successful. However, if the Gini coefficient is high, we have few AJE in TTC and there is not much room for reducing this number in ETTC.

α	10	15	20(E)	20(X)	20(U)
0.5	34239(1.27%)	76669(1.90%)	130893(1.09%)	24002(30.60%)	73923(2.19%)
0.55	34625(2.54%)	76311(1.21%)	125452(1.68%)	24602(28.20%)	71441(2.90%)
0.6	37589(1.67%)	80853(2.29%)	132242(2.00%)	21158(28.58%)	75873(4.53%)
0.65	34715(3.00%)	77448(2.84%)	129353(1.74%)	20773(27.20%)	76391(3.23%)
0.7	22429(4.47%)	52559(4.42%)	84902(6.21%)	15865 (8.50%)	53036(8.20%)

Table 13: AJE in FC&T

α	10	15	20(E)	20(X)	20(U)
0.5	33869(2.34%)	75841(1.34%)	129574(1.71%)	23799(31.18%)	73564(2.66%)
0.55	34232(3.70%)	75621(2.12%)	124390(2.51%)	24326(29.00%)	70319(4.43%)
0.6	37343(2.32%)	80233(3.04%)	131079(2.86%)	21120(28.71%)	75685(4.76%)
0.65	34471(3.69%)	76482(4.05%)	128848(2.13%)	20596(27.82%)	75695(4.11%)
0.7	22361(4.76%)	52376(4.75%)	84666(6.47%)	15830(8.7%)	52830(8.55%)

Table 14: AJE in C&T

α	10	15	20(E)	20(X)	20(U)
0.5	31329(13.51%)	66925(17.04%)	111088(16.81%)	15792(54.34%)	63078(15.00%)
0.55	31524(14.55%)	66776(16.84%)	106789(18.13%)	16414(59.10%)	60833(16.22%)
0.6	33408(17.26%)	69727(18.70%)	109950(19.71%)	13094(55.80%)	61750(20.00%)
0.65	32268(12.76%)	67031(18.59%)	110960(17.54%)	13187(53.79%)	63509(17.77%)
0.7	20293(17.49%)	44044(22.80%)	68009(27.59%)	11422(34.12%)	38858(28.04%)

Table 15: AJE in ETTC

α	20E	20X
0.5	23.3144	23.8314
0.7	15.972	16.9664

Table 16: t-test for comparison of ETTC and CT.

	TTC	FC&T	C&T	ETTC
1	25.85	25.92	25.87	25.84
1-2	45.55	46.33	45.82	45.82
1-3	65.73	65.85	66.17	65.99
1-4	84.93	85.68	85.54	85.22
1-5	100	100	100	100

(a) (20E,0.7)

	TTC	FC&T	C&T	ETTC
1	35.03	33.49	35.03	35.00
1-2	53.64	52.71	53.7	53.65
1-3	74.00	71.88	73.98	73.99
1-4	90.32	89.90	90.32	90.34
1-5	100	100	100	100

(b) (20E,0.6)

	TTC	FC&T	C&T	ETTC
1	40.23	38.39	40.23	40.22
1-2	60.09	59.14	60.1	60.01
1-3	78.85	76.82	78.83	78.82
1-4	91.72	91.07	91.72	91.73
1-5	100	100	100	100

(c) (20E,0.5)

Table 17: Equal quota environment (E) with different correlations

	TTC	FC&T	C&T	ETTC
1	45.15	45.15	45.15	45.12
1-2	74.63	74.63	74.67	74.61
1-3	82.55	82.54	82.8	82.54
1-4	92.6	92.59	93.11	93.2
1-5	100	100	100	100

(a) (20U,0.7)

	TTC	FC&T	C&T	ETTC
1	59.55	59.54	59.54	59.55
1-2	77.39	77.38	77.38	77.44
1-3	86.14	86.11	86.11	86.12
1-4	94.84	94.84	94.85	94.81
1-5	100	100	100	100

(b) (20U,0.6)

	TTC	FC&T	C&T	ETTC
1	64.73	64.73	64.74	64.75
1-2	80.25	80.25	80.24	80.28
1-3	88.92	88.9	88.91	88.96
1-4	95.29	95.29	95.28	95.32
1-5	100	100	100	100

(c) (20U,0.5)

Table 18: Unequal quota environment (U) with different correlations.

	TTC	FC&T	C&T	ETTC
1	77.6	77.59	77.59	77.6
1-2	85.08	85.07	85.91	85.88
1-3	91.38	91.38	91.67	91.38
1-4	96.08	96.17	96.54	96.16
1-5	100	100	100	100

(a) (20X,0.7)

	TTC	FC&T	C&T	ETTC
1	66.69	66.69	66.69	66.70
1-2	87.95	87.94	87.93	87.96
1-3	92.58	92.59	92.49	92.59
1-4	97.11	97.13	97.03	97.12
1-5	100	100	100	100

(b) (20X,0.6)

	TTC	FC&T	C&T	ETTC
1	63.95	63.92	63.93	63.96
1-2	87.83	87.57	87.57	89.33
1-3	94.62	94.32	94.32	96.13
1-4	97.42	97.11	97.1	98.92
1-5	100	100	100	100

(c) (20X,0.5)

Table 19: Extreme quota environment (X) with different correlations.

α	20E	20U	20X
0.7	0.5572	0.718	0.8875
0.6	0.6326	0.7948	0.8608
0.5	0.6772	0.8223	0.8558

(a) TTC

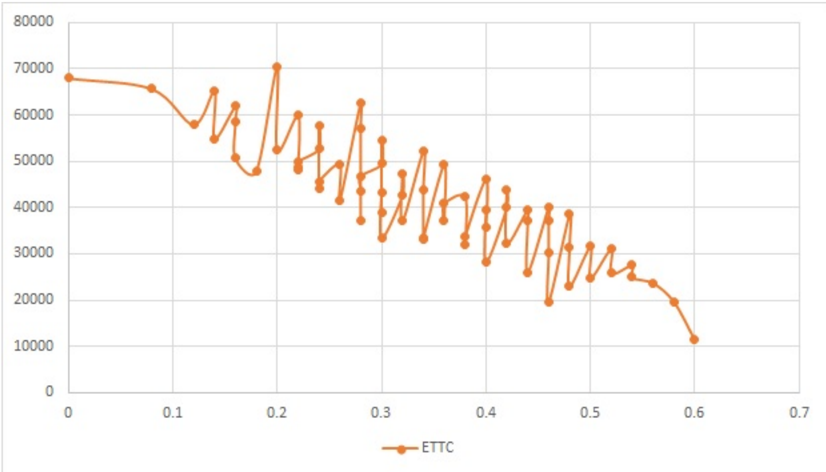
α	20E	20U	20X
0.7	0.5572	0.7372	0.8775
0.6	0.6326	0.7947	0.8608
0.5	0.6772	0.8229	0.8573

(b) CT

α	20E	20U	20X
0.7	0.5572	0.7012	0.8776
0.6	0.6325	0.7948	0.8609
0.5	0.6770	0.8233	0.8708

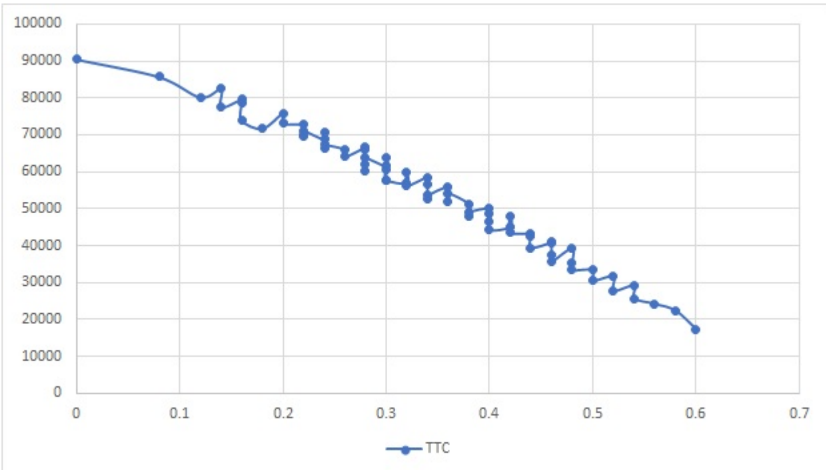
(c) ETTC

Table 20: Comparing efficiency for different quota configurations



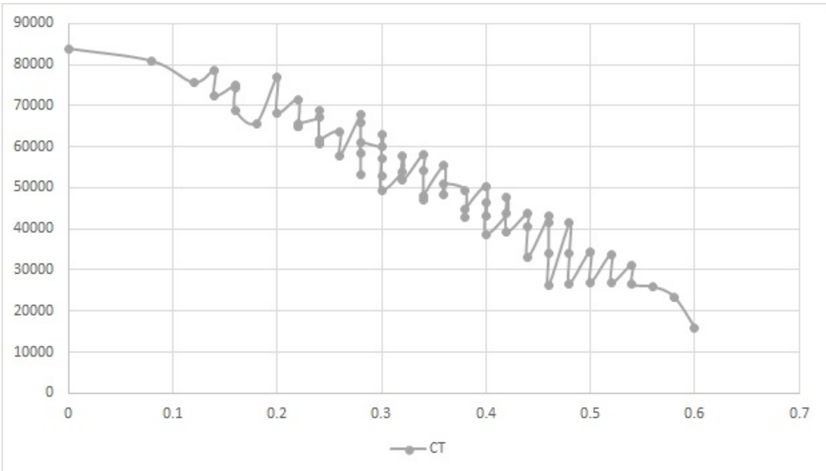
ETTC	Coef.	t	$p > \|t\|$
Slope	-84507.94	-14.08	0.000
Intercept.	70752.8	32.58	0.000

Table 21: Linear regression of ETTC with different Gini coefficients (Adj R-squared =0.7409)



TTC	Coef.	t	$p > \|t\|$
Slope	-129229.1	-56.32	0.000
Intercept.	98497	118.64	0.000

Table 22: Linear regression of TTC with different Gini coefficients (Adj R-squared = 0.9787)



CT	Coef.	t	$p > \|t\|$
Slope	-119040.4	-28.52	0.000
Intercept.	92174.86	61.90	0.000

Table 23: Linear regression of CT with different Gini coefficients (Adj R-squared = 0.9237)

Appendix to Chapter 4

We claim that more seats should be allocated to more favored schools to reduce justified envy. This is supported by the general intuition that this distribution will give us a better matching. Here we show results were the distribution of seats is the opposite, for comparison.

We test the unequal scenario, with two seats for the three top schools and seven seats for each of the two bottom schools ($20(U)'$), and an extreme scenario where 16 seats are given to the last preferred school in the common ranking and one seat for each of the remaining schools ($20(X)'$). The results for AJE can be seen in Tables 26-28. The tables demonstrate that if more seats are given to less favored schools then fewer students are matched to higher-ranked schools.

α	20(X)′	20(U)′	20(X)	20(U)
0.5	110375	145206	15792	63078
0.55	108615	139539	16414	60833
0.6	103325	140867	13094	61750
0.65	88314	125145	13187	63509
0.7	52313	78450	11422	38858

Table 24: AJE in TTC: case where more seats are given to low-ranked schools (shown by prime) vs. high-ranked schools

α	20(X)′	20(U)′	20(X)	20(U)
0.5	99334	138309	23799	73564
0.55	97727	132597	24326	70319
0.6	93488	134670	21120	75685
0.65	88314	125145	20596	75695
0.7	52313	78450	15830	52830

Table 25: AJE in C&T: case where more seats are given to low-ranked schools (shown by prime) vs. high-ranked schools

α	20(X)′	20(U)′	20(X)	20(U)
0.5	99334	138309	15792	63078
0.55	97727	132597	16414	60833
0.6	93488	134670	13094	61750
0.65	88314	125145	13187	63509
0.7	52313	78450	11422	38858

Table 26: AJE in ETTC: case where more seats are given to low-ranked schools (shown by prime) vs. high-ranked schools

Bibliography

[1] Atila Abdulkadiroğlu, Yeon-Koo Che, and Yosuke Yasuda. Resolving conflicting preferences in school choice: The "Boston Mechanism" reconsidered. *American Economic Review*, 101(1):399–410, 2011.

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

[3] Atila Abdulkadiroğlu, Yeon-Koo Che, Parag A Pathak, Alvin E Roth, and Olivier Tercieux. Minimizing justified envy in school choice: The design of New Orleans' OneApp. *Mimeo*, 2017.

[4] Jose Alcalde and Antonio Romero-Medina. Fair student placement. *Theory and Decision*, 83(2):293–307, 2017.

[5] Itai Ashlagi, Duncan S Gilchrist, Alvin E Roth, and Michael A Rees. NEAD chains in transplantation. *American Journal of Transplantation*, 11(12):2780–2781, 2011.

[6] Itai Ashlagi, Duncan S Gilchrist, Alvin E Roth, and Michael A Rees. Nonsimultaneous chains and dominos in kidney-paired donation revisited. *American Journal of Transplantation*, 11(5):984–994, 2011.

[7] Michel Balinski and Tayfun Sönmez. A tale of two mechanisms: Student placement. *Journal of Economic Theory*, 84(5):73–94, 1999.

[8] David Cantala and Szilvia Pápai. Reasonably and securely stable matching. *Mimeo*, 2014.

[9] Yan Chen and Onur Kesten. Chinese college admissions and school choice reforms: A theoretical analysis. *Journal of Political Economy*, 125(1):99–139, 2017.

[10] Umut Dur, Robert G Hammond, and Thayer Morrill. The secure Boston mechanism: Theory and experiments. *Experimental Economics*, forthcoming, 2018.

[11] Umut Dur and Utku Ünver. Two-sided matching via balanced exchange: Tuition and worker exchanges. *Mimeo*, 2015.

[12] Lars Ehlers and Thayer Morrill. (Il)legal assignments in school choice. *Mimeo*, 2017.

[13] Haluk I Ergin. Efficient resource allocation on the basis of priorities. *Econometrica*, 70(6):2489–2497, 2002.

[14] David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[15] Rustamdjan Hakimov and Onur Kesten. The equitable top trading cycles mechanism for school choice. *International Economic Review*, 59(4):2219–2258, 2018.

[16] Patrick Harless. From behind the veil: Evaluating allocation rules by ex-ante properties. *Mimeo*, 2015.

[17] Eun Jeong Heo. Preference profiles for efficiency, fairness, and consistency in school choice problems. *International Journal of Game Theory*, 48(1):243–266, 2019.

[18] Will Jones and Alexander Teytelboym. The international refugee match: A system that respects refugees' preferences and the priorities of states. *Refugee Survey Quarterly*, 36(2):84–109, 2017.

[19] Alexander Karpov. Preference diversity orderings. *Group Decision and Negotiation*, 26(4):753–774, 2017.

[20] Onur Kesten. Student placement to public schools in the US: two new solutions. *Mimeo*, 2004.

[21] Onur Kesten. On two competing mechanisms for priority-based allocation problems. *Journal of Economic Theory*, 127(1):155–171, 2006.

[22] Onur Kesten. School choice with consent. *The Quarterly Journal of Economics*, 125(3):1297–1348, 2010.

[23] Andrew Kloosterman and Peter Troyan. Efficient and essentially stable assignments. *Mimeo*, 2016.

[24] Taro Kumano. Strategy-proofness and stability of the Boston mechanism: an almost impossibility result. *Journal of Public Economics*, 105:23–29, 2013.

[25] Antonio Miralles. School choice: The case for the Boston mechanism. In *Auctions, market mechanisms and their applications*. Springer, 2009.

[26] Thayer Morrill. Making efficient school assignment fairer. *Mimeo*, 2013.

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

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

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

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

[31] Marek Pycia and Utku Ünver. Outside options in neutral allocation of discrete resources. *Mimeo*, 2016.

[32] Alvin E Roth. The economics of matching: Stability and incentives. *Mathematics of Operations Research*, 7(4):617–628, 1982.

[33] Alvin E Roth and Elliott Peranson. The redesign of the matching market for American physicians: Some engineering aspects of economic design. *American Economic Review*, 89(4):748–780, 1999.

[34] Alvin E Roth, Tayfun Sönmez, and Utku Ünver. Kidney exchange. *Quarterly Journal of Economics*, 119:457–488, 2004.

[35] Alvin E Roth and Marilda A Oliveira Sotomayor. *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*. Cambridge University Press, 1990.

[36] Mark A Satterthwaite and Hugo Sonnenschein. Strategy-proof allocation mechanisms at differentiable points. *Review of Economic Studies*, 48(4):587–597, 1981.

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

[38] Lars-Gunnar Svensson. Strategy-proof allocation of indivisible goods. *Social Choice and Welfare*, 16:557–567, 1999.

Appendix: Matlab codes

This appendix contains the main codes that are used in the simulations.

```
%--------------------------------------------------------------
%------ generating random preference and priority profiles-----
clear all
clc
% generates random (preference/priority) profiles
% 5 agents and 3 objects, 100 sample from preferences and priorities
objects=3;
agents=5;
number_of_preference_profiles=100;
number_of_priority_profiles=100;

for j=1:number_of_preference_profiles
agents_transpose=[];
for i=1:agents
%generate preference profile for one agent only [shown by "c"]
a=-1:objects;
a(a==0)=[];
b=perms(a); % all possible permutations
r = randi([1 size(b,1)],1,1) ;% random 1by1 matrix between 1 and size(b,1)
c=b(r,:);
agents_transpose=[agents_transpose; c];
agents_preference=agents_transpose';
end
agents_preference_p(:,:,j)=agents_preference
end

for j=1:number_of_priority_profiles
objects_transpose=[];
for i=1:objects
b=randperm(agents); % all possible permutations
objects_transpose=[objects_transpose; b];
objects_priority=objects_transpose';
end
objects_priority_p(:,:,j)=objects_priority
end


save('a5o3-1','objects_priority_p','agents_preference_p','agents','objects')
```

```
%----------------------------------------------------
%-----------TTC model -----------------------------

clear all
clc

% "-1" in preference profile means that agent does not want to participate
% in the market anymore, if an agent/ object is gone we will put zero in
% the profile
total_JE=0;
load('a5o3-32-1.mat')
q=1; % for saving final results in excel
 all_objects_quota_vector=[1 1 1]
for aa=1:size(agents_preference_p,3)
    for oo=1:size(preserved_profile,3)
quota=all_objects_quota_vector;


agents_preference=agents_preference_p(:,:,aa)
objects_priorities=preserved_profile(:,:,oo)


for i=1:2
agents_ID(i,:)=1:size(agents_preference_p,2);
end

for i=1:2
objects_ID(i,:)=1:size(preserved_profile,2);
end

%%%%%%%--------------------delete agents with "-1"-----------------
%%%%%%%-----------------------------------------------------------
% define matching matrix, second row is agents' ID and first row is alloted
% student
match=zeros(2,size(agents_ID,2)); %matching matrix
for i=1:size(agents_ID,2) %second row of matching matrix is students ID first row will↙
be filled with sschool allocation
    match(2,i)=i;
end

should_be_deleted_columns=[];
for i=1:size(agents_preference,2)
 if agents_preference(1,i)==-1
     should_be_deleted_columns=[should_be_deleted_columns i];
 end
end

% mus be deleted should be deleted columns with -1
for i=1:size(match,2)
    for j=1:length(should_be_deleted_columns)
```

```
        match(1,should_be_deleted_columns(j))=-1
    end
end

  agents_preference(:,should_be_deleted_columns)=[];


 %---------------find agents' ID :needed if we want to delete from priority table
 agents_index=[];
 for i=1:size(agents_ID,2)
     for j=1:size(should_be_deleted_columns,2)
   if agents_ID(1,i)==should_be_deleted_columns(j)
       agents_index=[agents_index agents_ID(2,i)];
   end
     end
 end




 agents_ID(:,should_be_deleted_columns)=[];
 agents_ID(1,:)=1:size(agents_ID,2); %delete from agents_preferencec

 for i=1:length(agents_index)
 objects_priorities(objects_priorities==agents_index(i))=0;

 end

[ objects_priorities ] = zeros_button( objects_priorities );




%-- if there are part of a chein the value becomes one. This is usuful to
%check cycle formation

object_in_cycle=zeros(1,size(objects_priorities,2));
agent_in_cycle=zeros(1,size(agents_preference,2));



%%%%%--------------------------end of delete agents with "-1" ------
%%%%%-------------------------------------------------------------



% "i" is the maximum numbers that we can have in the cycle
ia_c=1;
for i=1:100000000%size(agents_ID,2)  % not match completely with size
```

```
    %---------agents that have minus 1
    %-----------------------------------
    should_be_deleted_columns=[];
for i=1:size(agents_preference,2)
 if agents_preference(1,i)==-1
     should_be_deleted_columns=[should_be_deleted_columns i];
 end
end

  agents_preference(:,should_be_deleted_columns)=[];
agent_in_cycle(:,should_be_deleted_columns)=[];
agents_index=[];
 for i=1:size(agents_ID,2)
     for j=1:size(should_be_deleted_columns,2)
   if agents_ID(1,i)==should_be_deleted_columns(j)
       agents_index=[agents_index agents_ID(2,i)];
   end
     end
 end

for i=1:size(agents_index,2)
    for j=1:size(match,2)
   if match(2,j)==agents_index(i)
       match(1,j)=-1
   end
    end
end

 agents_ID(:,should_be_deleted_columns)=[];
 agents_ID(1,:)=1:size(agents_ID,2); %delete from agents_preferencec

 for i=1:length(agents_index)
 objects_priorities(objects_priorities==agents_index(i))=0;

 end

[ objects_priorities ] = zeros_button( objects_priorities );


if size(agents_ID,2)==0 % if there is no more agents in the market
    break
end

    %------------ end of agents that have minus one
    %----------------------------------------------

 if   agent_in_cycle(ia_c)~=1   %if there is a cycle then we go to else
 for j=1:size(objects_ID,2)
agent_in_cycle(ia_c)=1
io=agents_preference(1,ia_c)
```

```
io_c=relevent_c_sc(objects_ID,io)
object_in_cycle(io_c)=1
ia=objects_priorities(1,io_c)
ia_c=relevent_c_st(agents_ID,ia)

if   agent_in_cycle(ia_c)==1 % as soon as we realize that there is a cycle we start↙
allocation
    break
end

 end


 else

     new_matching_set=[]; %used in "reduced_quota_accordin_gmatching" function

     %start allocation
   for j=1:size(objects_ID,2)

     io_c=relevent_c_sc(objects_ID,io)
  %   all_objects_quota_vector(io_c)=all_objects_quota_vector(io_c)-1
     ia=objects_priorities(1,io_c)
     ia_c=relevent_c_st(agents_ID,ia)
     io=agents_preference(1,ia_c)
     match(1,ia)=io
     [new_matching_set] = reduced_quota_according_matching(io,ia,new_matching_set)
   end
   % new quota vector
   for i=1:size(new_matching_set,2)
    o= new_matching_set(1,i)
    o_c=relevent_c_sc(objects_ID,o)
    all_objects_quota_vector(o_c)=all_objects_quota_vector(o_c)-1

end
  % delete schools
   [objects_priorities,objects_ID,agents_preference]=delmatch_school1( match,↙
objects_ID,objects_priorities,agents_preference,all_objects_quota_vector);
   %op are not needed in simulation
   [agents_preference,agents_ID,objects_priorities]=delmatch_student1( match,agents_ID,↙
agents_preference,objects_priorities);

   all_objects_quota_vector(all_objects_quota_vector==0)=[]
   objects_ID(1,:)=1:size(objects_ID,2);
   agents_ID(1,:)=1:size(agents_ID,2);

   % -------------end of callinig elimination function--------------------
object_in_cycle=zeros(1,size(objects_ID,2));
agent_in_cycle=zeros(1,size(agents_ID,2));
```

```
%-------------------zeros on button function-------------------------
 agents_preference  = st_zeros_button( agents_preference );
 objects_priorities  = sc_zeros_button( objects_priorities );
%--------------------end of zeros on button function -----------------
ia_c=1; %after allocation we start from begining
end
if size(agents_ID,2)==0 % if there is no more agents in the market
    break
end

if size(all_objects_quota_vector,2)==0
    for i=1:size(match,2)
        if match(1,i)==0
            match(1,i)=-1;
        end

    end
    break
end


end
% we need original preference and priorities to be able to find blocking
% pairs
agents_preference=agents_preference_p(:,:,aa);
objects_priorities=objects_priority_p(:,:,oo);

[ blocking_pairs ] = block_finder( agents_preference,objects_priorities,match );

A(q,1)=aa;
A(q,2)=oo;
A(q,3)=blocking_pairs;
%A(q,4)=total_cycles(oo);
q=q+1;

clear agents_ID
clear  objects_ID
all_objects_quota_vector=quota;

total_JE=total_JE+blocking_pairs;
    end
end

output=xlswrite('C:\Users\pooya\Dropbox\mechanism design\matlab codes\mix - with↙
extentions\TTC-1-2-m\segmented_JE_32.xlsx',A);
```

```
%-------------Clinch and Trade -------------------
%-------------------------------------------------

clear all
        clc
        % we will generate sample filr in another profile
       % load('a20o5alpha0.7.mat')


load ('a20o5alpha0.5.mat')
%load ('studdent possible profiles')


q=1;
DP=zeros(size(student_preference_p,1),1);
total_blockings=0;

%sc=4;
%st=90;

       for sc=1:size(school_preference_p,3)

        for st=1:size(student_preference_p,3)



        student_preference=student_preference_p(:,:,st); %run for different samples
        school_preference=school_preference_p(:,:,sc);
                Original_sc_preference=school_preference; % needed for one loop in↙
future

%-------------- define student ID---------------------
        for i=1:2 %ID assigning to students  and schools
        student_ID(i,:)=1:size(student_preference,2);
        school_ID(i,:)=1:size(school_preference,2);
   end

             %%%%%%below is matching matrix
        match=zeros(2,size(student_ID,2)); %matching matrix
        for i=1:size(student_ID,2) %second row of matching matrix is students ID first↙
row will be filled with sschool allocation
            match(2,i)=i;
        end
        %%%%%%end of matching matrix

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  SChools
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  Capacity

        school_capacity=[2 2 2 7 7]; % can't be in sample file
        old_school_capacity=zeros(1,size(school_capacity,2)); % need to define↙
```

```
something for clinching process. if thehy are equal in clinching, the clinching loop↙
will end

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        school_in_cycle=zeros(1,size(school_preference,2));%if a school is tentatively↙
in cycle we will put a number of 1 for it. ?? do we need it for clinching process??
        student_in_cycle=zeros(1,size(student_preference,2));


         %"ia" is a generic element for school and "ibb" is a generic element for↙
student
        % k is the student we start with

        %need a benchmark : if school capacity is not changing in any clinching
        %process then we are done
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%clinching process%%%%%%%%%%%%
        for i=1:size(student_preference,2) % maximum all students can be assigned in↙
clinching process
        if old_school_capacity==school_capacity
            break
        else
            old_school_capacity=school_capacity; % we need it for clinching process(we↙
will continue the clinching process up until the point in which there is no more change↙
in school capacity)

        [ match, school_capacity] = FC( student_preference,school_preference,↙
school_capacity,match,student_ID,school_ID );


        %%%%%%%%%%%%%elimination after clinching process
        [school_preference,school_ID,student_preference]=delmatch_school1( match,↙
school_ID,school_preference,student_preference,school_capacity);
        [student_preference,student_ID,school_preference]=delmatch_student1( match,↙
student_ID,student_preference,school_preference);

        school_ID(1,:)=1:size(school_ID,2);
        %ia_c=school_ID(1,1)
        student_ID(1,:)=1:size(student_ID,2);
        % -------------end of callinig elimination function--------------------

        %-------------------zeros on button function-------------------------
         student_preference  = st_zeros_button( student_preference );
         school_preference  = sc_zeros_button( school_preference );
        end
        end

        %---------------end of clinching
         %----------------------------school capacities deletion
     del_cap=[]; %rows that should be deleted
```

```
     for i=1:size(school_capacity,2)
         if school_capacity(1,i)==0
             del_cap=[del_cap i];
         end
     end
     %---- ID should be deleted from students oreference in the followings
     %if ~isempty(del_cap) % otherwise there is no deletion
    % for i=1:length(del_cap) % deleting scool ID from students preference
      %for j=1:size(school_ID,2)
      % if  school_ID(1,j)==del_cap(i)
     %student_preference(student_preference==school_ID(2,j))=0;
      % end
     % end
    % end

     %------------------------------------------------------------

     school_capacity(:,del_cap)=[];
   % end
     %----------------------------end of deletion

        scc=nnz(school_capacity); % a condition for doing the rest(the process may end
in this loop) (there should be some remained capacities for schools)
        % IMP: " just in case where capacity of schools are equal to students
        % and no one wants to be unmatched
  if scc~=0
        ibb=1;
        ia=1; %just to define something
        %ibb_c=relevent_c_st(student_ID,ibb);
        for i=1:size(Original_sc_preference,2) % because school might be deleted in
clinching process
        if size(find(school_ID(2,:)==ia),2)==0
            ia=ia+1;
        end
        end


        ia_c=relevent_c_sc(school_ID,ia);
       %%%%%%need to delete stident and school in cycle
school_in_cycle=zeros(1,size(school_ID,2));
student_in_cycle=zeros(1,size(student_ID,2));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% starting cycles

    for m=1:size(match,2) %number of runs

        ibb_c=1;
    ua=nnz(match(1,:));%un allocated items
        if ua==size(match,2) % if all are allocated then we are done
```

```
            break
        end
    %student_in_cycle(ibb)=1;
    if length(student_ID)==0
        break
    end


    for i=1:size(student_ID,2)+1 %in each cycle we have to go through the loop and

    %check one more student and one more school
    % when we finished a complete cycle we will start allocations od that
    % cycle. the allocations will start tthrough "if" below
    %--------------------conditional starting of allocation-----------------
    if student_in_cycle(ibb_c)==1 % if this equality holds we need to start allocation
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        M=[]; %this is for clinching process after each trade (set of ID od schools↙
which have zero capacity after the trade)
        N=[]; % samae as above [N stans for students that need to form blocking pair]
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    for j=1:i
     ia=student_preference(1,ibb_c);  % we start allocation from stu
         ia_c=relevent_c_sc(school_ID,ia);

     if match(1,ibb)~=0 %if we already alocated two pairs we have to break the cycle
        break
    else
        match(1,ibb)=ia; %if its not over doing then do the allocation
        school_capacity(ia_c)=school_capacity(ia_c)-1;
        %we should find the schools with zero capacity
     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

     end
     %%% run clinch after the trade only for this students


    ibb=school_preference(1,ia_c); %this is not for allocation, we will do this just to↙
go one step ahead
    ibb_c=relevent_c_st(student_ID,ibb);
     if school_capacity(ia_c)==0
        m=school_ID(2,ia_c) ;
        M=[M m];
      end
    end

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%55


      [clinch_student_preference,clinch_student_ID,clinch_school_preference]↙
=delmatch_student1( match,student_ID,student_preference,school_preference);
```

```
[clinch_school_preference,clinch_school_ID,removed_student_preference] =↙
delmatch_school2( match,school_ID,clinch_school_preference,clinch_student_preference,↙
school_capacity );
         % We dont need to double count students that already have been
         % allocated in the cycle. for that reason, we first delete the
         % aloocated students and then we look for the students that need
         % to be clinched

del_cap=[]; %columns that should be deleted
     for i=1:size(school_capacity,2)
         if school_capacity(1,i)==0
             del_cap=[del_cap i];
         end
     end
     school_capacity(:,del_cap)=[];
      school_preference(:,del_cap)=[];
      keep_school_ID=school_ID;
      school_ID(:,del_cap)=[];
      school_ID(1,:)=1:size(school_ID,2);
     % Now we will find the ID of students who are pointing to them
     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    for i=1:size(clinch_student_ID,2)
         for j=1:length(M)
        if clinch_student_preference(1,i)==M(j)
            n=clinch_student_ID(2,i); % ID of students hat need to be in clinching↙
process
            N=[N n];
         end
         end
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
N_C=[];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%55
     for i=1:size(N,2)
N_C(i)=relevent_c_st(clinch_student_ID,N(i));
end
%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:length(M)
removed_student_preference(removed_student_preference==M(i))=0;
end
removed_student_preference=st_zeros_button(removed_student_preference);

hp_sc=[];
    %%%%%%%%%%% find this students highest preference
for s=1:size(N,2)
   hp_sc(s)=removed_student_preference(1,N_C(s));%the schools that are highest↙
preference of students
    % first ine is gone in the cycle, but we did not delete it yet ,the second one↙
should be the one we
    % are looking for is the one that can be used for clinching process
```

```
end
hp_sc_c=[];
for i=1:size(hp_sc,2)
    hp_sc_c(i)=relevent_c_sc(school_ID,hp_sc(i));
end
  %school_ID(:,del_cap)=[];

%hp_sc=unique(hp_sc); % avoid repeating elements
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sample_sans_rep=[]; % we may have double deduction in capacity, i.e if  hp_sc=[4 4];we
will fix thi by this function
old_sample_sans_rep=[]; % if there is a change in this then we have the allocation and
we reduce capacity of school by one unit

for j=1:size(hp_sc,2) % start a loop for clinching
 sample=[sample_sans_rep hp_sc(j)];
 sample_sans_rep=unique(sample);
 old_school_capacity=school_capacity;
% now check the schools column

% if schools first preference is equal to students ID then these two will be matched to
each other
for i=1:size(hp_sc,2)
        for k=1:school_capacity(hp_sc_c(i))
    if clinch_school_preference(k,hp_sc_c(i))==N(i)
        if match(1,N(i))==0
        match(1,N(i))=hp_sc(i);
        %if length(sample_sans_rep)~=length(old_sample_sans_rep)
          school_capacity(hp_sc_c(i))=school_capacity(hp_sc_c(i))-1;
        %end
    end
    end

    end
end
     old_sample_sans_rep= sample_sans_rep;

      if length(old_school_capacity)~=length(school_capacity) % for the next "if"
matrix dimensions must be same
        break
    end
    if old_school_capacity==school_capacity
        break
    end
end
 % end
    %schools and students that should be deleted
     %---------------------------school capacities deletion
```

```
     %---- ID should be deleted from students oreference in the followings
     if ~isempty(del_cap) % otherwise there is no deletion
     for i=1:length(del_cap) % deleting scool ID from students preference
      for j=1:size(school_ID,2)
       if  school_ID(1,j)==del_cap(i)
     student_preference(student_preference==keep_school_ID(2,j))=0;
       end
      end
     end

     %------------------------------------------------------------
del_cap=[]; %rows that should be deleted
     for i=1:size(school_capacity,2)
         if school_capacity(1,i)==0
             del_cap=[del_cap i];
         end
     end
     school_capacity(:,del_cap)=[];
     school_ID(:,del_cap)=[];
     school_preference(:,del_cap)=[];
    end
     %---------------------------end of deletion

    %--------------call elimination function-----------------
    [school_preference,school_ID,student_preference]=delmatch_school1( match,school_ID,↙
school_preference,student_preference,school_capacity);
    [student_preference,student_ID,school_preference]=delmatch_student1( match,↙
student_ID,student_preference,school_preference);
    school_ID(1,:)=1:size(school_ID,2);
    %ia_c=school_ID(1,1)
    student_ID(1,:)=1:size(student_ID,2);
    % -------------end of callinig elimination function--------------------
    school_in_cycle=zeros(1,size(school_preference,2));
    student_in_cycle=zeros(1,size(student_preference,2));
    %-------------------delete scshools from students preferences-------
    [ student_preference ] = delete_full_schools( school_ID,student_preference,↙
school_preference_p );
    %--------------------zeros on button function-------------------------
     student_preference  = st_zeros_button( student_preference );
     school_preference  = sc_zeros_button( school_preference );
    %---------------------end of zeros on button function -----------------
    break %after aloocating we have to stop the cycle of the first round
    else %if we dont have complete cycle, we will go one step a head
        student_in_cycle(ibb_c)=1; % if it is not in the loop then we put it in the↙
loop and henceforth it would be equal to "1"
        ia=student_preference(1,ibb_c);%first available preference of student "ia" is↙
the school that student "ibb" wants the most
        ia_c=relevent_c_sc(school_ID,ia);

    end
```

```
     ua=nnz(match(1,:));%un allocated
        if ua==size(match,2)
            break
        end

    if school_in_cycle(ia_c)==1
        M=[];
        N=[];
    for j=1:i %up to the point that cycle is complete. we did it up to step "i" so we
will allocate up to this part
    ibb=school_preference(1,ia_c); %we start from the point of repetition to allocate
    ibb_c=relevent_c_st(student_ID,ibb);
    ia=student_preference(1,ibb_c);%we should start to allocate from students
    ia_c=relevent_c_sc(school_ID,ia); %% ia has already been deleted in step j=1

    if match(1,ibb)~=0 %we may have more repetition than normal in some cases. so as
soon as we get in to loop we will cut the loop
        break
    else
        match(1,ibb)=ia;
        school_capacity(ia_c)=school_capacity(ia_c)-1;

        %%%%%%%%%%%%%%%%%%
        %%%%%%%%%%%%%%%%%%%%
        if school_capacity(ia_c)==0
           m=school_ID(2,ia_c) ;
        M=[M m];

     % find the ID of students who are pointing to them
     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%% if student is in schools one of high priorities then clinching
%%%%%%%%%% is done
%  for j=1:size(school_capacity,2)
%if
%end
 % end
%end
     end

    end

    end
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%55
 %%%%% after this allocation we should have deletion before doing the
 %%%%% clinching process

 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%55
 %%%%%%%%%start deletion
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%---------------------------school capacities deletion
        del_cap=[]; %rows that should be deleted
        for i=1:size(school_capacity,2)
            if school_capacity(1,i)==0
                del_cap=[del_cap i];
            end
        end
    %---- ID should be deleted from students oreference in the followings
    if ~isempty(del_cap) % otherwise there is no deletion
    for i=1:length(del_cap) % deleting scool ID from students preference
     for j=1:size(school_ID,2)
      if  school_ID(1,j)==del_cap(i)
    student_preference(student_preference==school_ID(2,j))=0;

    %school_ID(:,del_cap(i) )=[];

      end
     end
    end

    %-----------------------------------------------------------

    school_capacity(:,del_cap)=[];
    school_ID(:,del_cap)=[];
      school_ID(1,:)=1:size(school_ID,2);
    school_preference(:,del_cap)=[];
   end
    %---------------------------end of deletion
      [clinch_student_preference,clinch_student_ID,clinch_school_preference]↙
=delmatch_student1( match,student_ID,student_preference,school_preference);
[clinch_school_preference,clinch_school_ID,removed_student_preference] =↙
delmatch_school2( match,school_ID,clinch_school_preference,clinch_student_preference,↙
school_capacity );

      %  [clinch_student_preference,clinch_student_ID,clinch_school_preference]↙
=delmatch_student1( match,student_ID,student_preference,school_preference);
       % [clinch_school_preference,clinch_school_ID,clinch_student_preference]↙
=delmatch_school1( match,school_ID,school_preference,student_preference,↙
school_capacity);

        % We dont need to double count students that already have been
        % allocated in the cycle. for that reason, we first delete the
        % aloocated students and then we link for the students that need
        % to b eclinched
        del_cap=[]; %rows that should be deleted
    for i=1:size(school_capacity,2)
        if school_capacity(1,i)==0
            del_cap=[del_cap i];
        end
    end
```

```
     school_capacity(:,del_cap)=[];
          school_preference(:,del_cap)=[];
                keep_school_ID=school_ID;

               school_ID(:,del_cap)=[];
      school_ID(1,:)=1:size(school_ID,2);

N=[];
    for i=1:size(clinch_student_ID,2)
         for j=1:length(M)
        if clinch_student_preference(1,i)==0
            n=clinch_student_ID(2,i); % ID of students hat need to be in clinching
process
            N=[N n];
         end
         end
    end

    N_C=[];
    %%%%%%%%%%%%%%find students column%%%%%%%%%%%%%%%%%%
for i=1:size(N,2)
N_C(i)=relevent_c_st(clinch_student_ID,N(i));
end
%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:length(M)
removed_student_preference(removed_student_preference==M(i))=0;
end
removed_student_preference=st_zeros_button(removed_student_preference);

%%%%%%%%%%%%%%%%%%%%%%%%
hp_sc=[];
%%%%%%%%%%% "hp_sc" are the "schools" which has high preference for students
%%%%%%%%%%% who are supposed to do the clinching in this step
for s=1:size(N,2) %
   hp_sc(s)=removed_student_preference(1,N_C(s));% highest rpiority in school
    % first one is gone in the cycle, the second one should be the one we
    % are looking for is the one that can be used for clinching process
end

% now check the schools column
for i=1:size(hp_sc,2)
    hp_sc_c(i)=relevent_c_sc(school_ID,hp_sc(i));
end
%school_ID(:,del_cap)=[];
clinch_school_preference=sc_zeros_button( clinch_school_preference );

sample_sans_rep=[]; % we may have double deduction in capacity, i.e if  hp_sc=[4 4];we
will fix thi by this function
old_sample_sans_rep=[]; % if there is a change in this then we have the allocation and
we reduce capacity of school by one unit
```

```
for j=1:size(hp_sc,2) % start a loop for clinching
 sample=[sample_sans_rep hp_sc(j)];
 sample_sans_rep=unique(sample);
 old_school_capacity=school_capacity;
% now check the schools column

% if schools first preference is equal to students ID then these two will be matched to↙
each other
for i=1:size(hp_sc,2)
        for k=1:school_capacity(hp_sc_c(i))
    if clinch_school_preference(k,hp_sc_c(i))==N(i)
        if  match(1,N(i))==0
        match(1,N(i))=hp_sc(i);
          school_capacity(hp_sc_c(i))=school_capacity(hp_sc_c(i))-1;
        end
    end

    end
end
     old_sample_sans_rep= sample_sans_rep;

      if length(old_school_capacity)~=length(school_capacity) % for the next "if"↙
matrix dimensions must be same
        break
    end
    if old_school_capacity==school_capacity
        break
    end
end
%---------------------------school capacities deletion
         del_cap=[]; %rows that should be deleted
         for i=1:size(school_capacity,2)
             if school_capacity(1,i)==0
                 del_cap=[del_cap i];
             end
         end
     %---- ID should be deleted from students oreference in the followings
     if ~isempty(del_cap) % otherwise there is no deletion
     for i=1:length(del_cap) % deleting scool ID from students preference
      for j=1:size(school_ID,2)
       if  school_ID(1,j)==del_cap(i)
     student_preference(student_preference==keep_school_ID(2,j))=0;

       end
      end
     end

     %----deleting school ID
      % if school_capacity(ia_c)==0
```

```
     %   school_ID(:,ia_c)=[];
    %school_ID(1,:)=1:size(school_ID,2);
    %school_capacity(:,ia_c)=[];
    %end

     %-------------------------------------------------------------

     school_capacity(:,del_cap)=[];
     school_ID(:,del_cap)=[];
       school_ID(1,:)=1:size(school_ID,2);
     school_preference(:,del_cap)=[];
    end
     %---------------------------end of deletion
    % when allocation of first round ends, all the in cycles should be equal to
    % zero

    %students and schools that should be deleted

    %--------------call elimination function-----------------
    [school_preference,school_ID,student_preference]=delmatch_school1( match,school_ID,↙
school_preference,student_preference,school_capacity );
    [student_preference,student_ID,school_preference]=delmatch_student1( match,↙
student_ID,student_preference,school_preference);
    school_ID(1,:)=1:size(school_ID,2);
    student_ID(1,:)=1:size(student_ID,2);
    % -------------end of callinig elimination function--------------------
    school_in_cycle=zeros(1,size(school_preference,2));
    student_in_cycle=zeros(1,size(student_preference,2));
    %--------------------delete scshools from students preferences-------
    [ student_preference ] = delete_full_schools( school_ID,student_preference,↙
school_preference_p );
    %--------------------zeros on button function-------------------------
     student_preference  = st_zeros_button( student_preference );
     school_preference  = sc_zeros_button( school_preference );
    %---------------------end of zeros on button function -----------------
    break  %for breaking the biggest loop, eliminating alocations and re-run the loop
    else
      school_in_cycle(ia_c)=1; %because we start from school in each rounf of the loop,↙
we may head back to same thing
      ibb=school_preference(1,ia_c);
      ibb_c=relevent_c_st(student_ID,ibb);
    end
    %%%%%%%%%%%%%%%%%%%%end of allocation%%%%%%%%%%

    end

    end

        end
```

```
    %-----l ast end is for doint TTC if clinching process does not work-----%   end
     student_preference=student_preference_p(:,:,st);
     school_preference=school_preference_p(:,:,sc);
    [ blocking_pairs ] = block_finder( student_preference,school_preference,match );
    total_blockings=total_blockings+blocking_pairs;
    clear school_ID
    clear student_ID

    % save for comparing individual profiles
A(q,1)=sc;
A(q,2)=st;
A(q,3)=blocking_pairs;
q=q+1;
% end of saving
     student_preference=student_preference_p(:,:,st);

[ new_DP ] = distribution_function( match,student_preference);
DP=DP+new_DP;

         end
       end

       DP_av=DP/(st*sc)


       output=xlswrite('C:\Users\pooya\Dropbox\mechanism design\matlab codes\mix - with↙
extentions\C&T\output.xlsx',A);
```

```
%--------------- Equitable top trading cycle --------------------
%----------------------------------------------------------------

clear all
clc

load('a10o5alpha0.65.mat')
DP=zeros(size(student_preference_p,1),1);
total_blockings=0;
q=1;

st=33;
student_preference=student_preference_p(:,:,st);

for i=1:2
student_ID(i,:)=1:size(student_preference,2); %define 2 rows for student id, in futre↙
the first row would be column ID
end

sc=20;
school_preference=school_preference_p(:,:,sc); %sc=1


%school_capacity=xlsread('C:\Users\pooya\Dropbox\economic design\matlab codes\ETTC\TTC.↙
xlsx',3);
school_capacity=[2 2 2 2 2];
original_school_capacity=school_capacity;
for i=1:2
school_ID(i,:)=1:size(school_preference,2);
end
school_in_cycle=zeros(1,size(school_preference,2));%if a school is tentatively in cycle↙
we will put a number of 1 for it.
student_in_cycle=zeros(1,size(student_preference,2));
%below is matching matrix
match=zeros(2,size(student_ID,2)); %matching matrix
for i=1:size(student_ID,2) %second row of matching matrix is students ID first row will↙
be filled with sschool allocation
    match(2,i)=i;
end
%%%%%%end of matching matrix

[ ETTC_pairs ] = ETTC_pair_v2( school_preference,school_capacity ) % fiinding school-↙
student pairs
nn = nnz(ETTC_pairs); %number of none zero elements (maximum number of loops)

% ETTC_pairs are set of doable matches fro schoool preference
ETTC_pairs(:,:,2)=0; %each element would be equal to one if they are in cycle

ia_c=1; % a generic element for scool that we are working on it
    ii=1; % number of row that is going to be matched
```

```
    ibb_r=1;
mm=1;
zz=1;


for jj=1:50*nn% until every thing is allocated



    if ETTC_pairs(ii,ia_c,2)==1 % when a loop forms

        % ia_c=1;
         %ii=1;
         ibb=ETTC_pairs(ii,ia_c); % student ID
         ETTC_pairs(:,:,2)=0;

    % start allocating--------------------------------------------------
    for m=1:mm


%%%%%%%%%%%%%%%%%%%%%%%%%%%for second round (student column)

ibb_c=find_column_st( student_ID,ibb ); % columns that have students that need to be
allocated
ia_c_i=ia_c; %??
ia=student_preference(1,ibb_c); %first preference of student (that is the column that w
should go for it)


match(1,ibb)=ia;

for i=1:size(ETTC_pairs,1) % find the student that has been matched
if ETTC_pairs(i,ia_c)==ibb
ii=i;
end
end
ETTC_pairs(ii,ia_c,2)=1;


ia_c=find_column_st( school_ID,ia );

ibb= highest_rank( ETTC_pairs,school_preference,ia_c,ia_c_i );


    end
    %-------end of allocation

[ school_capacity,school_preference,school_ID,student_preference,S_CO,s,ETTC_pairs ] =
new_capacity_v2( match,original_school_capacity,school_preference,school_ID,
student_preference,ETTC_pairs );%school capacity reduced from matching matrix
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% delete assigned students in "ETTC_pairs"

for i=1:size(match,2)
    if match(1,i)~=0
        ETTC_pairs(ETTC_pairs==match(2,i))=0;

    end
end
ETTC_pairs=st_zeros_button(ETTC_pairs);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
%%%%%%%%%%%%%%%%%%%%%%%%end of deleting assgned students

%%%%%%%%%%%%%%%%%this should be corrected
[ ETTC_pairs ] = ETTC_pair_v3( school_preference,school_capacity,ETTC_pairs );
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    % delete
    mm=1;
[ student_ID,student_preference ] = del_assigned_st( match,student_ID,↙
student_preference,s );
         ETTC_pairs(:,:,2)=0;


[ student_preference ] = st_zeros_button( student_preference );
irs );
ii=1;
ia_c=1;
    else

if mm==1
      ETTC_pairs(1,ia_c,2)=1; % first element that participated
ibb=ETTC_pairs(1,ia_c); % student ID of the first element

  else
ETTC_pairs(ibb_r,ia_c,2)=1; % next element that participated
ibb=ETTC_pairs(ibb_r,ia_c); % student ID of next element
  end
  %!!!! end of this consideration
ia_c_i=ia_c;
%%%%%%%%%%%%%%%%%%%%%%%%%%%for second round (student column)
student_column=[];
ibb_c=find_column_st( student_ID,ibb );
%%%%%%%%%%%%%%%%%end of it
ia=student_preference(1,ibb_c); %#first preference of student (that is the column that↙
w should go for it)
% we may have many items and we have to find bests of them
% we can have many students in  the school side
%Find column of school
```

```
school_column=[];
ia_c=find_column_st( school_ID,ia );
%%%%%%%%%%%% find the best among many
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
%%%%%%%%%%%%%%%we need to put monover here
if ia_c_i~=ia_c
ibb = highest_rank( ETTC_pairs,school_preference,ia_c,ia_c_i );
end

for i=1:size(ETTC_pairs,1) %best ranked agent must participate
if ETTC_pairs(i,ia_c)==ibb
ii=i;
end
end


mm=mm+1;
[ ibb_r ] = find_row_st( ETTC_pairs,ibb,ia_c );
    end
if isempty(ETTC_pairs)
    break
end
end
```

```
%--------finding Justified envy instances-----
%---------------------------------------------
function [ blocking_pairs ] = block_finder( student_preference,school_preference,match↙
)
n=0; %number of blocking pairs

for i=1:size(student_preference,2) % "i" stands for student
sc_m=match(1,i); % school which is matched
%-------for finding rank of the matched school in students preference -----
for j=1:size(student_preference,1)
   if student_preference(j,i)==sc_m
    rank_sc=j;
    break
   end
end
if rank_sc~=1 % if it is first rank we do not have blocking pair for sure
%-----finding the schools above the matchings with higher preferences---
    for j=1:rank_sc-1
sc=student_preference(j,i); % each school above the ranking

rank_st=[];
for k=1:size(match,2) %finding matched student with any of the schools above preference↙
??????we may have more than one student
if match(1,k)==sc
    st=match(2,k)
    for k=1:size(school_preference,1) %for finding rank of the matched thing in schools↙
preference list
   if school_preference(k,sc)==st
    rank_st=[rank_st k]
    break
end
end
end
end
%------------------
% here we  will ahve number of blocking pairs but we can have number of
% matched pairs that have been blocked.
%------------------

rank_st=max(rank_st); % if its blocked pairs we have to compare it with every number in↙
the set

if rank_st~=1
for k=1:rank_st-1
    if school_preference(k,sc)==i
        n=n+1;
    end
end
end
```

```
end
end
end

blocking_pairs=n;
end
```