

Making Efficient School Assignment Fairer

Thayer Morrill[†]

November, 2013

Abstract

Surprisingly, nearly every school district that utilizes a centralized assignment procedure makes a Pareto inefficient assignment. The two objections to the standard efficient algorithm, Top Trading Cycles (TTC), are that it makes unfair assignments and that it allows students to trade school-specific priorities. This paper identifies two shortcomings of TTC: students may trade even after they are guaranteed their top choice and the students who trade are chosen myopically. We introduce algorithms that correct these flaws and demonstrate through simulations that this makes the resulting assignment significantly fairer and effectively eliminates the trading of school-specific priorities.

Key Words: Top Trading Cycles, School Choice, Assignment.

JEL Classification: C78, D61, D78, I20

When choosing an assignment mechanism a school board must balance the following three priorities: strategyproofness, Pareto efficiency, and fairness. Efficiency and fairness are incompatible in the sense that there may not always exist an assignment that is both fair and efficient (Roth 1982). Strategyproofness is also incompatible with efficiency and fairness as there does not exist a strategyproof mechanism that always selects a fair and efficient assignment even when one exists (Kesten 2010). Fortunately, there are well known algorithms that

[†]North Carolina State University. I am grateful to Tayfun Sönmez for his suggestions. I would also like to thank Umut Dur, Robert Hammond, and Melinda Morrill. Email address: thayer_morrill@ncsu.edu.

satisfy any two of the three priorities. Typically, a school board faces a decision between Gale’s Top Trading Cycles algorithm (hereafter TTC), which is strategyproof and efficient, and Gale and Shapley’s Deferred Acceptance algorithm, which is strategyproof and fair.

From an economic standpoint, the most fundamental of these conditions is Pareto efficiency. Schools are a public resource, and it is clearly undesirable to allocate a public resource in a Pareto inefficient manner. And yet, this is what nearly every school district has chosen to do.¹ When choosing between an efficient assignment algorithm and a fair assignment algorithm, virtually every school district has chosen Deferred Acceptance, the fair assignment algorithm.² The only school district we know of that has implemented TTC is New Orleans.³

Boston was the first city to consider implementing these algorithms, and indeed, TTC was their initial recommendation. The following passage from the Boston Public Schools Student Assignment Task Force describes their thought process when deciding between the Deferred Acceptance algorithm and TTC (emphasis is theirs):⁴

The Gayle [sic] Shapley algorithm is driven by priorities only, which cuts down on the amount of choice afforded to families. The Top Trading Cycles algorithm also takes into account priorities while leaving some room for choice. Since choice was very important to many families who attended the community forums, we believe that having an assignment algorithm that leaves some room for choice is best. Further, neither Gayle [sic] Shapley or Top Trading Cycles can be manipulated by parental gaming. We also believe that the Top Trading algorithm will enable the BPS to ensure that the 50% walk preference remains sacred throughout all the assignment rounds. *Therefore, we strongly recommend that the BPS adopt the Top Trading assignment algorithm.*

¹As an example of the inefficiency, Abdulkadiroğlu et al. (2009) found that in the 2006-2007 assignment of eighth graders in New York City, more than 4,000 students could have been assigned to a more preferred school without harming any student.

²Cities that have adopted a version of the student-proposing Deferred Acceptance algorithm include New York City (Abdulkadiroğlu et al. 2005b, 2009), Boston (Abdulkadiroğlu 2005a), and Chicago (Pathak and Sönmez, forthcoming). Denver will begin using Deferred Acceptance in 2012. Recently, Deferred Acceptance has been adopted by all local authorities in England (Pathak and Sönmez, forthcoming).

³San Francisco voted to implement TTC, but they have not made the algorithm they use public.

⁴See page 15 of the Student Assignment Task Force, submitted to Boston School Committee on September 22, 2004. It is available at <http://www.bostonpublicschools.com/assignment>.

Although the task force strongly recommended TTC, the school board ultimately chose to implement Deferred Acceptance. Interestingly, the stated reason for this change was not a concern over fairness but rather that some priorities are school specific, such as having a sibling attend the same school or a student being able to walk to a school, and therefore should not be tradeable. In his memo to the Boston School Committee discussing the choice of algorithm, Superintendent Payzant writes: ⁵

Another algorithm we have considered, Top Trading Cycles Mechanism, presents the opportunity for the priority for one student at a given school to be “traded” for the priority of a student at another school, assuming each student has listed the other’s school as a higher choice than the one to which he/she would have been assigned. There may be advantages to this approach, particularly if two lesser choices can be “traded” for two higher choices. It may be argued, however, that certain priorities – e.g., sibling priority – apply only to students for particular schools and should not be traded away

It is well known that TTC is strategyproof, efficient, and unfair. What is not known is whether or not there exists an algorithm that is strategyproof, efficient, and fairer⁶ than TTC. Similarly, if a school board wishes to restrict the trade of certain priorities, is it possible to reduce the occurrence of these undesirable trades? This paper provides a positive answer to both questions. We introduce a strategyproof and efficient algorithm that is significantly fairer than TTC. Moreover, we show that we may effectively eliminate trading of “restricted” priorities. Since school boards clearly place a great emphasis on both fairness and limiting the trading of school-specific priorities, the hope is that the our algorithm will make school boards more inclined to adapt a Pareto efficient assignment algorithm.

TTC was introduced in Shapley and Scarf (1974) to find a competitive solution for the Shapley-Scarf housing market. In this model, each student is endowed with a house. The algorithm proceeds as follows. Each student points to her favorite house, and each house points to its owner. As there are a finite number of students, there must exist a cycle. In

⁵This memo can be found at <http://www.iipsc.org/resources/tpayzant-memo-05.25.2005.pdf>. Accessed August 6, 2013.

⁶When we say an assignment is fairer, we mean that it has fewer instances of justified envy.

each cycle, assign the student to the house she is pointing to, remove the students and houses assigned, and continue. This algorithm has many desirable features. It is Pareto efficient, strategyproof, individually rational, and makes the unique assignment that is in the core of the assignment problem (Roth and Postlewaite, 1977).

The school assignment problem differs from the house assignment problem in two important ways: schools may be assigned to more than one student, and no student ‘owns’ a school. Nevertheless, TTC generalizes to the school assignment problem in a natural way.⁷ Each student points to her most preferred school. Each school points to the student with highest priority. For each cycle, assign the student to the school she is pointing to, remove the student, and reduce the capacity of the school by one. When a school has no remaining capacity, remove the school. This process is repeated until all students are assigned or there are no remaining schools.

The differences between the school assignment and housing allocation problems are critical and give us an opportunity to improve the performance of TTC. The trading cycles, which make TTC efficient, are also what make TTC unfair. Each student in a cycle has highest priority at the school pointing at her, but she is assigned the school she is pointing to. Since her priority at her assignment is not considered, assigning her to that school is potentially unfair. This is how we are able to improve TTC in terms of fairness. First, we will demonstrate that TTC makes more trades than is necessary. Second, we demonstrate that these trades can be made in a more sophisticated way and thereby reduce the chances that the trade results in a violation of fairness.

Our first innovation is to reduce the number of trades we utilize to make an assignment. A reasonable restriction on an assignment algorithm is that if student i ’s favorite school is a , a has capacity for q_a students, and i is one of the q_a highest ranked students at a , then i is assigned a . This condition is called *mutual best*.⁸ We define a student to be *guaranteed* a spot at a if i has one of the q_a highest priorities at a . TTC satisfies mutual best; if i is guaranteed a spot at her most preferred school a , then i is always assigned to a . However, unless i has the highest priority at a , TTC allows her to trade her priority at a different school

⁷This generalization of TTC was introduced in Abdulkadiroğlu and Sönmez (2003).

⁸Note that not all algorithms used in practice satisfy mutual best. For example, a serial dictatorship and linear programming procedure do not satisfy mutual best. See Morrill (2012a) and Morrill (2012b) for discussions on how mutual best relates to Deferred Acceptance and TTC, respectively.

in order to be assigned to a . This trade is not necessary for efficiency, strategyproofness, or mutual best, and as a result, it introduces potentially unnecessary violations of fairness. Our first innovation is to introduce an algorithm, *Clinch and Trade* (C&T), that avoids these unnecessary trades.

Our second innovation is to have each school point in a more sophisticated manner. In any algorithm that satisfies mutual best, the student with highest priority at a school a cannot be assigned to a school she finds worse than a . However, it does not imply that this student must be satisfied before any of the other students who are also guaranteed a seat at a . In particular, a school with capacity q does not need to point to the student with highest priority but rather must point to one of the q -highest-priority students. Pointing to the student with highest priority is myopic as this student does not necessarily have high priority at other schools. Since the school priorities are known to the mechanism designer, a better approach is to point to the student that is most likely to have high priority at the school she points to. We call this *Prioritized Trading Cycles* (PTC). We demonstrate that PTC is strategyproof and efficient. Our final, and preferred, algorithm combines the two innovations. We call this *Prioritized Clinch and Trade* (PC&T).

A key advantage of PC&T is that we may control which of the students trade their priorities. For example, in Boston Public Schools if a child's sibling attends a school, then she is given highest priority at that school. However, the school board objected to allowing students to trade sibling priorities. We are able to easily modify PC&T in the following way to address this concern. Suppose school A has a capacity of q but the school board does not want to allow $q' < q$ of the students to trade their priorities. We call these priorities *restricted priorities*. Instead of pointing to the student with the highest average ranking at other schools among the q highest priority students, we simply point to the highest ranked student among the $q - q'$ students that have unrestricted priorities. We show that this effectively eliminates the trading of restricted priorities; however, this reduction comes at a cost to fairness.

We next explore which of the four algorithms is the most fair. Given that two algorithms are unfair, we interpret algorithm A to be *fairer* than algorithm B for a given assignment

problem if on average there are fewer instances of justified envy under A than under B.⁹ We run a wide variety of simulations to compare the relative fairness of the four algorithms, and we consistently find that PC&T is the fairest followed by PTC, and then Clinch and Trade. TTC is consistently the least fair among the algorithms and often by a wide margin.

Our paper is a contribution to the recent literature on school choice pioneered by Abdulkadiroğlu and Sönmez (2003). Abdulkadiroğlu (2013) is an excellent survey on the literature. Our paper is similar in spirit to Erdil and Ergin (2008) and Abdulkadiroğlu, Pathak, and Roth (2009). These papers demonstrate that a seemingly innocuous feature of the deferred acceptance algorithm, how ties in a school’s priority ranking are broken, can have a significant impact on the efficiency of deferred acceptance. Our paper also contributes to the literature on alternatives to TTC for making a Pareto efficient assignment. Papai (2000) introduces and characterizes a generalization of TTC called hierarchical exchange rules. Pycia and Unver (2010) characterize the set of Pareto efficient and group strategyproof mechanisms by generalization hierarchical exchange rules to a class of rules called trading cycles.

Most similar to our paper is Kesten (2004). Kesten (2004) introduces several important and innovative algorithms including the Efficiency Adjusted Deferred Acceptance Algorithm later described in Kesten (2010). Most relevant to the current paper is his algorithm Equitable Top Trading Cycles (ETTC) which addresses the same question that we do; is it possible to make TTC fairer? Roughly speaking, ETTC proceeds as follows. If school a has q_a available spots, then each of the top q_a -ranked students at a are allocated a seat at a . The algorithm considers student-school pairs (i, a) where i has been allocated a seat at a . Each pair (i, a) points to the unique pair (j, b) such that b is i ’s favorite school and j has the highest priority at a among students that have been allocated b . There must exist at least one cycle, and the algorithm assigns each student in a cycle to its favorite available school. There are a number of important additional details for which the reader should refer to the paper. For example, a student i may appear in multiple cycles or even the same cycle multiple times. All cycles are processed, but as i is only assigned one copy of her favorite school, Kesten defines an inheritance procedure for the “extra” copies. Similarly, after a student i is assigned, there is an inheritance procedure for the schools that i was allocated but i did not use in a trade.

⁹There are, of course, many alternative ways one could define ‘fairer’. A particularly strong condition would be that algorithm A is fairer than algorithm B if A is never unfair whenever B is fair. Under this definition, none of the four algorithms are comparable.

ETTC is strategyproof and Pareto efficient.

ETTC and our algorithms were developed independently and consequently take different approaches; however, both algorithms address the same shortcomings of TTC. In Kesten’s definition of ETTC, if i ’s favorite school is a and i has been allocated a seat at a , then all student-seat pairs involving i point to (i, a) . This serves much of the same role as our clinching procedure, and in particular, a student does not trade her priority at a different school in order to be assigned to a school she was initially guaranteed to be admitted to. A key difference is that C&T is able to iterate the clinching procedure whereas the inheritance rules of ETTC wait until each person initially allocated a school has been assigned before proceeding with the inheritance. Kesten’s algorithm is also strategic in its pointing process unlike TTC which is myopic. The fact that each student-school pair (i, a) points to the student with highest priority at school a means that ETTC handles cycles of length two perfectly. If (i, a) and (j, b) form a cycle in ETTC, then among students guaranteed a seat at a , j is the student least likely to cause justified envy and *vice versa*. In this way, ETTC is superior at handling cycles of length two than PC&T as PC&T considers a student’s priority at all schools. However, for cycles of length greater than two, it is better to consider a student’s ranking at all schools as this additional information is relevant.

1 Model and Algorithms

We consider a finite set of students $I = \{1, \dots, n\}$ and a finite set of schools $S = \{a, b, c, \dots\}$. Each school a has a capacity for $q_a \geq 1$ many students. Each student $i \in I$ has a complete, irreflexive, and transitive preference relation P_i over $S \cup \{\emptyset\}$. \emptyset represents a student being unassigned, and $q_\emptyset = \infty$. aP_ib indicates that i strictly prefers school a to b . Given P_i , we define the symmetric extension R_i by aR_ib if and only if aP_ib or $a = b$.

Each school $a \in S$ has a complete, irreflexive, and transitive priority rule \succ_a over I . In particular, $i \succ_a j$ is interpreted as student i has a higher priority for school a than student j . We define \succeq analogously to our definition of R . An **assignment** is a function $\mu : I \rightarrow S \cup \{\emptyset\}$ such that for each $a \in S$, $|\{i \in I \mid \mu(i) = a\}| \leq q_a$. An assignment is **Pareto efficient** if there does not exist another assignment ν such that $\nu(i)R_i\mu(i)$ for every $i \in I$ and $\nu(i)P_i\mu(i)$

for some i . An assignment μ is **fair** if there does not exist a student i and a school a such that $aP_i\mu(i)$ and $i \succ_a j$ for some j such that $\mu(j) = a$.

Given a school a with quota q_a , let G_a be the q_a -highest-ranked students at a . We say a student in G_a is **guaranteed** a spot at a . An **assignment mechanism** ϕ is a function from a preference profile of student and priority profile of schools to an assignment. A mechanism ϕ is **strategyproof** if reporting true preferences is each student's weakly dominant strategy.

Abdulkadiroğlu and Sönmez (2003) give a detailed descriptions of TTC. Given strict preferences of students and strict priority lists for schools, TTC assigns students to schools according to the following algorithm. In each round, each student points to her most preferred remaining school, and each school with available capacity points to the remaining student with highest priority. As there are a finite number of students, there must exist a cycle $\{o_1, i_1, \dots, o_K, i_K\}$ such that each school, o_j , and student, i_j , points to i_j and o_{j+1} , respectively (with $o_{K+1} \equiv o_1$). For each cycle, student i_j is assigned to school o_{j+1} , i_j is removed, and the capacity of o_{j+1} is reduced by one. When a school has no remaining capacity, it is removed.

TTC is efficient, because each student in a cycle receives her most preferred school among those that have already been assigned. Therefore, we cannot improve a student's assignment without harming a student that has been previously assigned. If there is a path from a school a to a student i , then i has the ability to be assigned to a by pointing at a . TTC is strategyproof for two reasons. First, a student's report does not effect which student a school points to. Second, once there exists a path from a school a to a student i , that path remains until i is assigned. When i is assigned, she receives her most preferred school among those with available capacity, so by revealed preference, she must weakly prefer her assignment to a . Therefore, she never does better than by pointing to her favorite school.

In particular, so long as we preserve paths once they exist and do not use student preferences to determine which student each school points to, then our algorithm will continue to be strategyproof.¹⁰

Prioritized Trading Cycles (PTC)

¹⁰For example, if each school points to the student with highest priority at her most preferred school, then the algorithm would no longer be strategyproof.

Round 1:

Have each student point to her favorite school. Have each school a point to the student in G_a with the highest average rank¹¹ at schools other than a . There must exist a cycle. For each cycle, assign the student the school she is pointing to and remove her. Reduce the quota of each school in a cycle by one, and remove any school that now has a capacity of zero.

Round i :

Have each student point to her favorite school. If the student a school was pointing to in round $i - 1$ has not yet been assigned, have the school continue to point to the same student. Otherwise, have each school a point to the student in G_a with the highest average rank at schools other than a . Assign any student in a cycle to the school she is pointing to, and remove students and reduce quotas accordingly.

The algorithm continues until all students have been assigned or no school has available capacity. We make no claim that pointing to the student with highest average rank is the optimal decision rule. In fact, one can imagine alternatives that might be superior. For example, if school a is typically over demanded and school b is typically under demanded, then we should weight a 's priorities more heavily and perhaps not consider b 's priorities at all. The optimal decision rule is outside the scope of this paper, but we demonstrate that the *ad-hoc* rule of taking the highest average ranking at other schools significantly outperforms simply pointing at the student with highest priority.

Proposition 1. *PTC is efficient, strategyproof, and satisfies mutual best.*

Proof. We prove something more general. Consider any decision rule to determine which student a school points to that does not depend on the preferences of the students. We prove the following. So long as the algorithm has each school continue to point to the same student until that student is assigned, and so long as each school a always points to a student in G_a , then the algorithm will satisfy mutual best, strategyproofness, and efficiency.

¹¹We refer to “highest average rank” for expositional convenience. In practice, we utilize a more sophisticated procedure. First, when calculating the average, we do not consider a student's rank at schools she is guaranteed to be admitted to as she can not cause justified envy at such a school. Second, we weight the ranking at schools by the available capacity of the school with schools with fewer seats weighted more strongly than schools with larger capacity. Note that the procedure is strategyproof and efficient for any decision rule used to select one of the guaranteed students.

Consider any student i and school a such that a is i 's most preferred school and $i \in G_a$. At most one school can be assigned to a in a given round. Since a always points to a student in G_a , then if a student not in G_a is assigned to a , in the same cycle a student in G_a must be assigned to a school other than a . Therefore, if i is not assigned to a in round k , then $i \in G_a$ in round $k + 1$. As a result, a can only reach full capacity, $G_a = \emptyset$, after i is assigned. Since a always points to a student in G_a , eventually a must point to i , and therefore, i is assigned to a .

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

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

A second shortcoming of TTC is it allows students to trade in order to receive their favorite school even if they are guaranteed admission to that school. These trades are unnecessary from an efficiency point of view and expose the algorithm to fairness violations. We illustrate this with Example 1.

Example 1. Suppose there are three students $\{i, j, k\}$ and two schools $\{a, b\}$. School a has a capacity of two while school b has a capacity of one. Define R and \succ according to the following rank-order lists:

R_i	R_j	R_k	\succ_a	\succ_b
b	a	b	i	j
a	b	a	j	k
			k	i

In the first round of TTC, $\{i, b, j, a\}$ form a cycle. Therefore, $TTC(R, \succ)$ assigns i , j , and k to b , a , and a respectively.

In Example 1, j has one of the two highest priorities at her most preferred school which has a capacity of two. Therefore, TTC will always assign j to a regardless of i and k 's preferences or j 's priority at b . However, TTC allows j to make an unnecessary trade with i . This trade causes a distortion. Compare TTC's assignment to assigning i , j , and k to a , a and b , respectively. This assignment respects priorities whereas the TTC assignment does not. The only student that does not get her top choice is i , and i has lower priority at her top choice, b , than the student assigned to b , k .

This motivates a new algorithm:

Clinch and Trade

Round 1:

- 1a** For each $i \in I$, if i is one of the q_a highest ranked student at i 's most preferred school a , then assign i to a , remove i and set $q_a = q_a - 1$. Now iterate until no student has one of the q_a highest priorities at her most preferred school a .
- 1b** Have each student that remains point to her most preferred school that has capacity greater than zero. Have each school with available capacity point to the highest ranked

student. Note that there must exist a cycle. For every cycle that exists, assign the student to the school she is pointing to, remove the student, and reduce the capacity of the school by one.

Round k :

k.a If the school i was pointing to in Round $k - 1$ still has available capacity, then i continues to point to the same school.

k.b Iteratively clinch until no student who was not pointing at the end of Round $k - 1$ has one of the q_a highest priorities at her most preferred school a .

k.c Have each remaining student point to her most preferred school with available capacity. Have each school with available capacity point to the highest ranked student. Again, there must exist a cycle. For every cycle that exists, assign the student to the school she is pointing to, remove the student, and reduce the capacity of the school by one.

Proposition 2. *Clinch and Trade is efficient, strategyproof, and satisfies mutual best.*

Proof. The proof of Proposition 2 is analogous to the proof of Proposition 1. Since every student assigned receives her favorite among unassigned schools, the algorithm is Pareto efficient. We do not let a student that is pointing at a school clinch that school. Therefore, the clinching process does not effect any existing path. In particular, once there exists a path from a school a to a student i , then that path remains until i is assigned. Therefore, i has no incentive to misrepresent her preferences and point to any school other than her most preferred school. Similarly, a student has no incentive to clinch a school unless it is her most preferred school. She can only clinch a school a if she is guaranteed admissions to a , and once a student is guaranteed admissions to a school, then she never loses that status. This also implies that Clinch and Trade satisfies mutual best as a student is never assigned a school worse than a school she is guaranteed admissions to. For a more formal argument, see Morrill (2012c).

□

Morrill (2012c) describes a number of properties of Clinch and Trade. The algorithm is Pareto efficient and strategyproof; however, it loses two desirable features of TTC. In particular, Clinch and Trade is bossy¹² and depends on the order in which cycles are removed. The Deferred Acceptance algorithm is also bossy, but TTC is not.

Our preferred algorithm combines PTC and Clinch and Trade. As in Clinch and Trade, we iteratively clinch assignments until no student is guaranteed admission to her most preferred school. We then determine trades by using the pointing process used in PTC. Each student points to her most preferred school. If the student a school was pointing to in round $i - 1$ has not yet been assigned, have the school continue to point to the same student. Otherwise, each school a points to the student in G_a with the highest average rank at schools other than a . We call this **Prioritized Clinch and Trade** (PC&T).

Corollary 1. *Prioritized Clinch and Trade is strategyproof, efficient, and satisfies mutual best.*

Proof. The proof of Proposition 2 holds for any pointing function so long as 1) a school always points to a student that is guaranteed admissions, 2) the pointing function does not depend on the preferences of the students and (3) once a school points to a student, it continues to point to the student until the student is assigned. Since PC&T satisfies all of these conditions, it is strategyproof, efficient, and satisfies mutual best. \square

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

¹²An assignment mechanism is bossy if a student may change the assignment of another student without changing her own assignment.

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

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

Note that TTC is not fair when there are two schools and the schools have sufficient capacity. We show via simulations that the intuition from Proposition 3 holds more generally. When there are fewer schools, average priority becomes a better predictor of a student's priority at her favorite school. While PC&T consistently outperforms TTC, the difference is most dramatic when there are relatively few schools with relatively large capacities.

Corollary 2. *If there are two schools and the total capacity of the two schools is greater than or equal to the number of students, then PC&T corresponds exactly to the student-proposing Deferred Acceptance algorithm.*

Proof. As is well known, the assignment generated by the student-proposing Deferred Acceptance algorithm Pareto dominates any other fair assignment. Since the assignment made by PC&T is efficient and fair, it must correspond to the Deferred Acceptance assignment. \square

Table 1 - Baseline Scenario

	TTC	C&T	PTC	PC&T
Instances of Justified Envy	88.88(0.30)	84.43(0.29)	66.81(0.23)	61.59(0.22)
Students with Justified Envy	9.49(0.03)	9.47(0.03)	8.58(0.02)	8.42(0.02)
Avg. Number Envied	9.34(0.02)	8.91(0.02)	7.83(0.02)	7.35(0.02)
Students Justifiable Envied	28.83(0.05)	27.26(0.05)	25.62(0.05)	24.10(0.05)
Avg. Number Envied By	3.00(0.01)	3.02(0.01)	2.54(0.01)	2.48(0.01)

This table considers several ways of evaluating the fairness of the four algorithms. In each scenario, there are 100 students, 5 schools, and each school has a capacity for 20 students. Preferences and priorities were uniformly drawn, and we ran 20,000 simulations. In the first row, we report the average number of instances of justified envy for each algorithm. If a student has justified envy, the third row reports the average number of students that she justifiably envies. The fourth row gives the average number of students that are justifiably envied by some student. The fifth row gives the average number of students that justifiably envy such a student.

2 Comparing the Algorithms Using Simulations

The three new algorithms introduced in the previous section are not unambiguously fairer than TTC. That is to say, there exist assignment problems where TTC is fair but C&T, PTC, and PC&T are not fair. The algorithms process students in different orders, and each time a student is assigned, a new subproblem is induced. Therefore, it is straightforward to generate an example where TTC never reaches a subproblem where it makes an unfair assignment but each of the other algorithms does. However, we provide evidence in this section that on average, each of the new algorithms has significantly fewer violations of fairness than TTC.

We ran a large number of simulations over a variety of assignment problems. Each time, we count the number of times a student has justified envy of another student. Strikingly, there was a consistent pattern among the new algorithms. In nearly every scenario we ran, the ranking of the algorithms from best to worst was PC&T, PTC, Clinch and Trade, and finally TTC. The only exception to this that we have found is when there is a large amount of excess capacity. In this scenario, Clinch and Trade can be fairer than PTC. The reason Clinch and Trade does particularly well in this scenario is the excess capacity means that a greater percentage of students are assigned through the iterative clinching process. Note that PC&T still performs the best of the four algorithms in this scenario.

Our baseline simulation model consists of 100 students, five schools, and capacities for each school of 20. This was chosen to be consistent with Abdulkadiroğlu et al. (2012). We ran 20,000 simulations with preferences and priorities uniformly drawn. Table 1 summarizes the performance of the four algorithms for a variety of ways of defining fairness. Our preferred method is to count the total instances of justified envy. The reason is that it matters both how many unfair assignments we make and how unfair these assignments are. For example, if a student’s assignment creates justified envy, it is better if only one student justifiably envies her than if five students justifiably envy her. Alternatively, we show the average number of students that have justified envy and the average number of students that create justified envy.

Table 1 demonstrates that our modifications have two impacts. First, they reduce the number of students that have justified envy. Second, when the algorithm does assign a student in a way that creates justified envy, there are fewer students that envy the assignment than under TTC. The combined result of these two improvements, is that the algorithms significantly reduce the number of instances of justified envy. For example, PTC had, on average, roughly 30% fewer violations of fairness.

Figure 1 gives the performance of the three new algorithms relative to TTC when the baseline model is perturbed in a number of ways. Each plot shows the average number of instances of justified envy as a percentage of TTC’s average number of fairness violations. In the top-right figure, we keep the capacity at all schools constant but vary the number of schools. In the bottom-left figure, we keep the number of students and schools constant but vary the capacity of each school. In the bottom-right figure, we vary the degree to which school priorities are correlated.

In all simulations, the three new algorithms consistently outperformed TTC. As Figure 1 indicates, there are several plausible scenarios where PC&T dramatically outperforms TTC. For example, when there are only two schools and sufficient total capacity for all students, PC&T is fair (Proposition 3). Note that TTC is not fair in this situation.

It is intuitive that we see the most dramatic improvement of PC&T (and PTC) relative to TTC when there are few schools. In these algorithms, among the set of students who are guaranteed a spot at a school, the school points to the student with the highest average rank

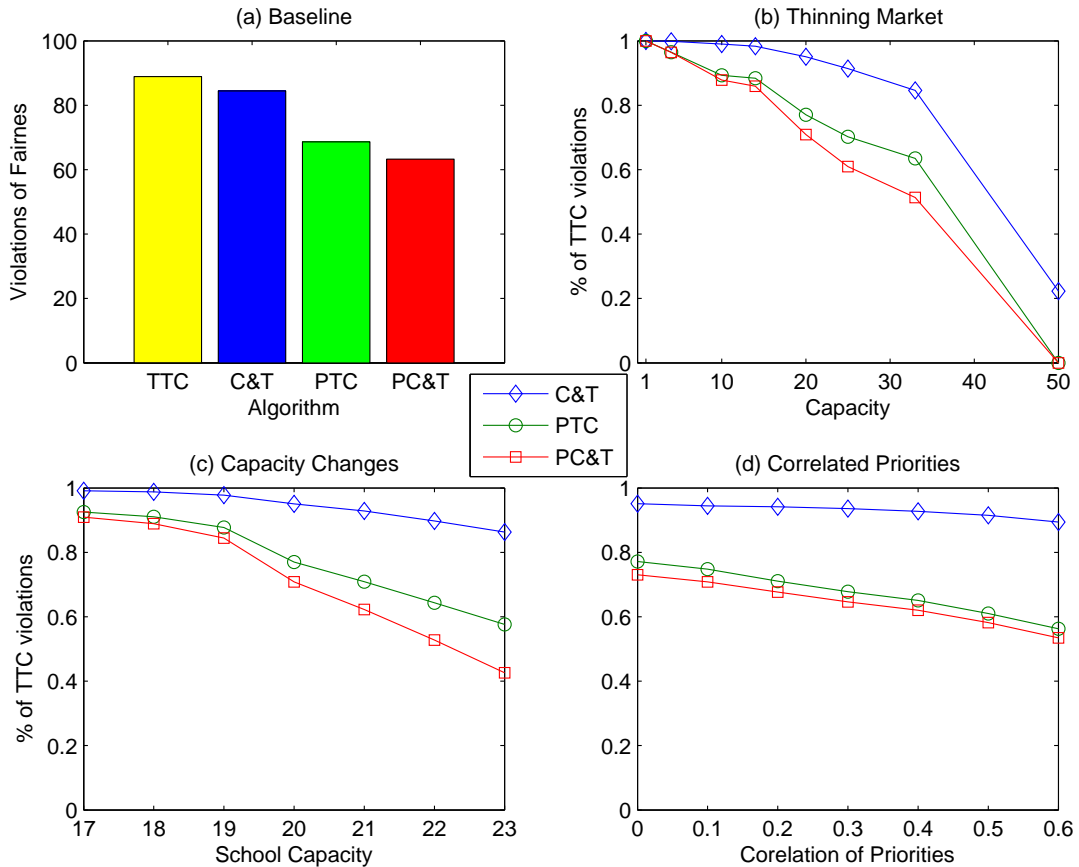


Figure 1: In each scenario, we ran 20,000 simulations.

(a) Baseline - 100 students and 5 schools, each with a capacity for 20 students. This figure gives the average number of instances of justified envy for each algorithm.

(b) Thinning Market - Here we vary the number of schools but keep the total capacity of all schools constant at 100. Specifically, we run simulations when there are 50 schools each with a capacity of 2, 25 schools each with a capacity of 4, 20 schools each with a capacity of 5, etc. The x-axis gives the capacity of the schools for a particular scenario. The y-axis is the each axioms average fairness violations as a percentage of TTC fairness violations.

(c) Capacity Changes - Here we keep the number of schools constant at 5, but we vary the capacity of each school. We start with a shortage of seats and increase until there is an excess number of total seats. Specifically, we consider the cases where each school has a capacity for 17 through 23 students.

(d) Correlated Priorities - Here we consider the scenario where students preferences over the schools are correlated. Specifically, for every student we specify one common ranking over schools, and for each student we draw (uniformly) an individual ranking over schools. For a given scenario, a students ranking over the schools is the convex combination of the two: $\alpha \cdot individual + (1 - \alpha) \cdot common$ where α varies from 0 to 1.

at the other schools. The fewer other schools there are, the better the average ranking is as an estimator of the students rank at the school she ultimately points to.

PC&T also significantly outperforms TTC when there is excess capacity. This is due to the iterative clinching process. The greater the overall capacity is, the more students have a chance to clinch a seat. The more students that clinch seats, the fewer trades PC&T makes relative to TTC. As the trades are where potential fairness distortions are introduced, in this environment PC&T significantly outperforms TTC.

We provide tables in the Appendix for the average number of fairness violations for each algorithm in each scenario.

3 Restricting Tradeable Priorities

In this section, we show how PC&T (and analogously PTC) may be easily modified if a school board wishes to limit the trading of certain priorities. As mentioned in the introduction, the primary objection to TTC by the school board in Boston was not fairness but rather students being able to trade certain priorities. For example, the highest priority at a school is given to students whose sibling attends that school, and the school board objected to these priorities being tradeable. Similarly, if the intention of walk-zone priorities are to either foster neighborhood school or reduce transportation costs, then a school board would find trading these priorities undesirable. Note that TTC is particularly vulnerable to these trades as students with a sibling that attend the school or that live within the walk zone are typically given the highest priority at a school, and under TTC, the students with the highest priorities are the ones most likely to be involved in a trade.

The intent of PC&T is to have each school point to the student least likely to create justified envy at the school she is assigned. But in general, PC&T can accommodate any decision rule and retain the same properties of strategyproofness and efficiency. In particular, suppose a school a has capacity q_a and let S_a be the q_a highest ranked students. Further, suppose that a subset of these students, $R_a \subset S_a$ have priorities that a school board does not want traded. We call R_a the set of restricted students and $S_a \setminus R_a$ the set of unrestricted students at a .

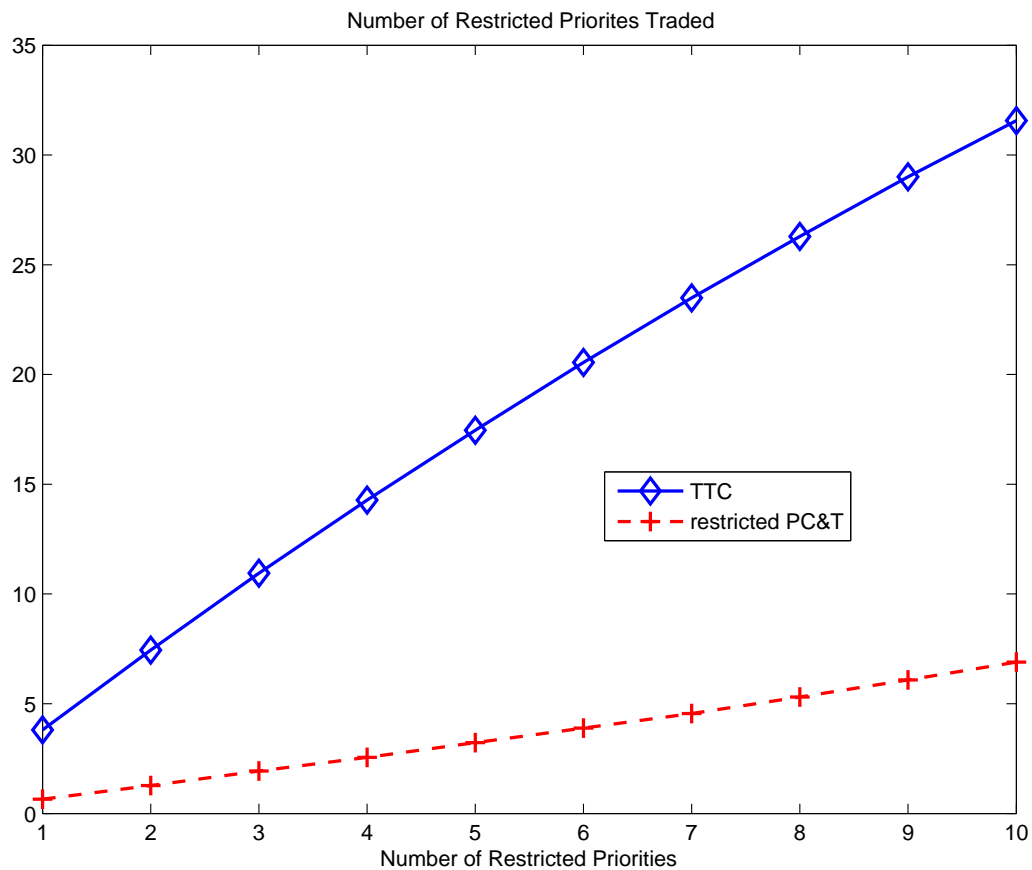


Figure 2: Average number of restricted priorities that are traded by TTC and PC&T. In each scenario, there are 100 students and 5 schools with a capacity for 20 students. The restricted priorities are always the highest priorities.

Then instead of pointing to the student in S_a least likely to create justified envy, we point to the unrestricted student least likely to create justified envy. We only point to a restricted student if there are no remaining unrestricted students. We call this modification *restricted Prioritized Clinch and Trade*.

By adjusting the pointing procedure in this manner, we are able to effectively eliminate the trading of restricted priorities. Figure 2 summarizes the impact of restricting priorities on both TTC and restricted PT&C. In each scenario, there are 100 students, 5 schools, and each school has a capacity for 20 students. We ran ten different scenarios. In each, the top x priorities at a school are restricted where x varies from one to ten. For each scenario, we ran 20,000 simulations. In each scenario, restricted PT&C reduces the number of restricted priorities that are traded by approximately 80%.

When determining who a school should point to in the standard PT&C, we consider all students who are guaranteed admission to a school. In restricted PT&C, we consider a subset of students. Therefore, we expect restricted PT&C to have more instances of justified

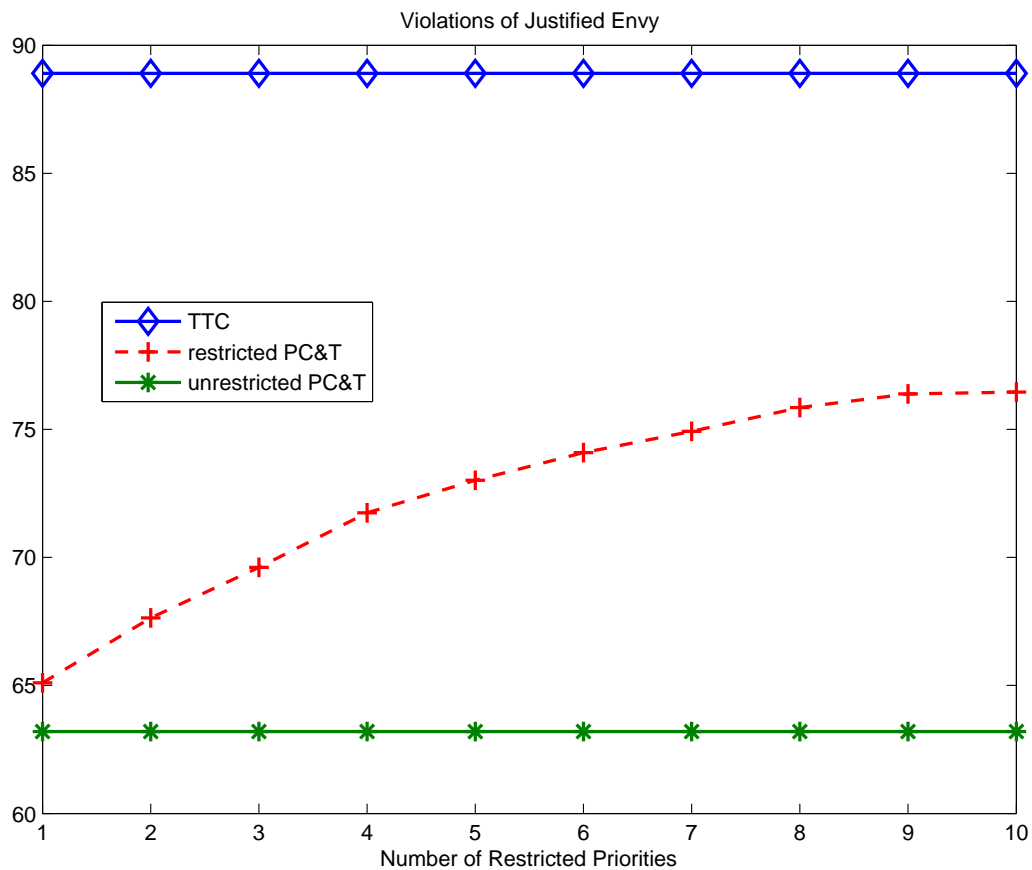


Figure 3: Average instances of justified envy when priorities are restricted. In each scenario, there are 100 students and 5 schools with a capacity for 20 students. The restricted priorities are always the highest priorities.

envy than PT&C. Figure 3 compares the average number of instances of justified envy for restricted PT&C compared to TTC and PT&C for the same simulations scenarios as in Figure 2. As predicted, restricted PT&C is significantly fairer than TTC but less fair than PT&C.

4 Conclusion

Abdulkadiroğlu and Sönmez (2003) demonstrate that the Top Trading Cycles is a strategyproof and efficient algorithm for assigning students to schools. However, nearly every school district has chosen the fair algorithm, Deferred Acceptance, over the efficient algorithm, Top Trading Cycles. Although it is impossible for an algorithm to be strategyproof, fair, and efficient, we demonstrate that it is possible to make Top Trading Cycles significantly fairer. This is achieved by reducing the number of trades that Top Trading Cycles makes and choosing the trades that it does make in a more strategic manner. As it is clearly

undesirable to assign students to schools in a Pareto inefficient manner, hopefully by making Top Trading Cycles fairer, more school districts will implement an efficient assignment algorithm.

References

- ABDULKADIROGLU, A., AND T. SONMEZ (2003): “School choice: A mechanism design approach,” *The American Economic Review*, 93(3), 729–747.
- ABDULKADIROGLU, A., C. Y., AND Y. YASUDA (2012): “Expanding “Choice” in School Choice,” *mimeo*.
- ABDULKADIROGLU, A. PATHAK, P. R. A., AND T. SONMEZ (2005a): “The Boston public school match,” *The American Economic Review*, 95, 368–371.
- (2005b): “The New York high school match,” *The American Economic Review*, 95, 364–367.
- (2009): “Strategyproofness versus Efficiency in Matching with Indifferences: Redesigning the NYC High School Match,” *The American Economic Review*, 99, 1954–1978.
- GALE, D., AND L. SHAPLEY (1962): “College admissions and the stability of marriage,” *American Mathematical Monthly*, 69(1), 9–15.
- KESTEN, O. (2004): “Student Placement to Public Schools in US: Two New Solutions,” *mimeo*.
- (2010): “School choice with consent,” *The Quarterly Journal of Economics*, 125(3), 1297–1348.
- MORRILL, T. (2012a): “An alternative characterization of the deferred acceptance algorithm,” *International Journal of Game Theory*.
- (2012b): “An Alternative Characterization of Top Trading Cycles,” *Economic Theory*.
- (2013): “Two Simple Variations of Top Trading Cycles,” *mimeo*.

PATHAK, P. A., AND T. SÖNMEZ (2012): “School Admissions Reform in Chicago and England: Comparing Mechanisms by Their Vulnerability to Manipulation,” *The American Economic Review*.

ROTH, A. (1982): “The economics of matching: Stability and incentives,” *Mathematics of Operations Research*, pp. 617–628.

ROTH, A., AND A. POSTLEWAITE (1977): “Weak versus strong domination in a market with indivisible goods,” *Journal of Mathematical Economics*, 4(2), 131–137.

ROTH, A., AND M. SOTOMAYOR (1992): *Two-sided matching: A study in game-theoretic modeling and analysis*. Cambridge Univ Pr.

5 Appendix

Table 2 - Response to Market Thickness

Schools	Cap/School	TTC	C&T	PTC	PC&T
100	1	151.47(0.21)	151.47(0.21)	151.47(0.21)	151.47(0.21)
25	4	152.68(0.27)	152.45(0.27)	147.25(0.26)	147.24(0.26)
10	10	130.76(0.31)	129.44(0.31)	116.78(0.28)	114.84(0.28)
7	14	166.26(0.33)	163.53(0.32)	147.00(0.28)	142.87(0.28)
5	20	89.02(0.30)	84.62(0.28)	68.57(0.24)	63.12(0.23)
4	25	71.35(0.27)	65.23(0.25)	50.09(0.20)	43.49(0.19)
3	33	61.09(0.24)	51.70(0.20)	38.79(0.15)	31.39(0.13)
2	50	19.18(0.14)	4.27(0.04)	0.00(0.00)	0.00(0.00)

This table is the summary data for scenario (b) - **Thinning Markets** in Figure 1 on Page 17. We compare the number of fairness violations when there are many objects with small capacities to when there are few objects with large capacities. In each scenario, there are 100 students. We vary the number of schools and quota at each school, but in each scenario all schools have the same quota and the total number of available seats is 100 (except when there are 7 and 3 schools where integer constraints result in the total market capacity being 98 and 99, respectively). For each scenario, we ran 20,000 simulations.

Table 3 - Varying School Capacity

Quota	TTC	C&T	PTC	PC&T
17	449.11(0.48)	445.16(0.48)	415.57(0.45)	408.42(0.45)
18	331.39(0.41)	327.40(0.40)	301.73(0.36)	294.60(0.36)
19	206.98(0.34)	202.40(0.33)	181.53(0.28)	174.80(0.28)
20	88.53(0.29)	84.17(0.28)	68.15(0.23)	62.72(0.22)
21	50.20(0.24)	46.61(0.22)	35.58(0.18)	31.26(0.17)
22	30.10(0.19)	27.00(0.17)	19.36(0.13)	15.88(0.12)
23	18.39(0.15)	15.87(0.13)	10.60(0.10)	7.84(0.08)

This table is the summary data for scenario (c) - **Capacity Changes** in Figure 1 on Page 17. In each scenario, there were 100 students and 5 schools. The table shows the affect of varying the capacity of each school. For each scenario, we ran 20,000 simulations.

Table 4 - Correlated Priorities

Alpha	TTC	C&T	PTC	PC&T
0.0	88.78(0.30)	84.42(0.28)	68.51(0.24)	64.82(0.23)
0.1	123.71(0.37)	116.86(0.35)	92.50(0.29)	87.62(0.28)
0.2	203.65(0.44)	191.72(0.42)	144.76(0.35)	137.79(0.35)
0.3	302.09(0.51)	282.78(0.49)	204.75(0.43)	195.30(0.43)
0.4	394.88(0.59)	366.33(0.59)	256.88(0.51)	244.85(0.52)
0.5	456.71(0.71)	417.83(0.72)	278.62(0.59)	265.78(0.61)
0.6	436.86(0.82)	390.59(0.83)	245.86(0.61)	233.45(0.62)

This table is the summary data for scenario (d) - **Correlated Priorities** in Figure 1 on Page 17. This table shows the affect of correlated rankings. Specifically, for every student we specify one common ranking over schools, and for each student we draw (uniformly) an individual ranking over schools. For a given scenario, a students ranking over the schools is the convex combination of the two: $\alpha \cdot individual + (1 - \alpha) \cdot common$ where α varies from 0 to .6. For each scenario, we ran 20,000 simulations.