Economic Theory manuscript No.
(will be inserted by the editor)

Two Simple Variations of Top Trading Cycles

Thayer Morrill

May 2014

Abstract Top Trading Cycles is widely regarded as the preferred method of assigning students to schools when the designer values efficiency over fairness. However, Top Trading Cycles has an undesirable feature when objects may be assigned to more than one agent as is the case in the school choice problem. If agent *i*'s most preferred object *a* has a capacity of q_a, and *i* has one of the q_a highest priorities at *a*, then Top Trading Cycles will always assign *i* to *a*. However, until *i* has the highest priority at *a*, Top Trading Cycles allows *i* to trade her priority at other objects in order to receive *a*. Such a trade is not necessary for *i*'s assignment and may cause a distortion in the fairness of the assignment. We introduce two simple variations of Top Trading Cycles in order to mitigate this problem. The first, Clinch and Trade, reduces the number of unnecessary trades but is bossy and depends on the order in which cycles are processed. The second, First Clinch and Trade, is nonbossy and independent of the order in which cycles are processed but allows more unnecessary trades than is required to be strategyproof and efficient. Both rules are strategyproof.
JEL Classification: C78, D61, D78, I20

Keywords Top Trading Cycles · School Choice · Assignment

This paper is concerned with the assignment of agents to discrete resources when monetary transfers are prohibited. In particular, we focus on the application of assigning students to public schools. As Abdulkadiroglu and Sönmez (2003) detail in their seminal paper on school assignment, the two mechanisms widely considered by economists are Gale and Shapley's Deferred Acceptance algorithm (Gale and Shapley 1962) and Gale's Top Trading Cycles (Shapley and Scarf 1974), hereafter TTC. In this paper, we introduce two alternatives to TTC. Both mechanisms are variations of TTC designed to mitigate an undesirable feature of TTC when objects may be assigned to more than one agent.

North Carolina State University. I would like to thank Justin Burkett, Umut Dur, Lars Ehlers and Melinda Morrill for their helpful comments. Email address: thayer_morrill@ncsu.edu.

North Carolina State University

When choosing an assignment mechanism a school board aims to balance the following three priorities: strategyproofness, efficiency, and fairness. An assignment is *fair* if there is no student i and school a such that i strictly prefers a to her assignment and i is ranked higher at a than one of the students assigned to a.[1] Efficiency and fairness are incompatible in the sense that there may not always exist an assignment that is both fair and efficient (Balinski and Sönmez, 1999 and 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). Therefore, the market designer must prioritize between the conditions. If, for example, she values strategyproofness first, fairness second, and efficiency third, then the deferred acceptance algorithm should be her choice. If she values efficiency first, fairness second, and strategyproofness third, then she should run the efficiency-adjusted deferred acceptance algorithm introduced by Kesten (2010).

This paper seeks to address the question of how to assign students when the designer values strategyproofness first, efficiency second, and fairness third.[2] Traditionally, the recommendation would be to run TTC (Abdulkadiroglu and Sönmez 2003). Indeed, when the capacity of each object assigned is only one, TTC may be thought of as the most fair among strategyproof and efficient mechanisms (Morrill 2013). However, we demonstrate that this is not necessarily the case for the school assignment problem.

TTC for the school assignment problem, as introduced in Abdulkadiroglu and Sönmez (2003), proceeds as follows. Each student points to her favorite school, and each school points to the student with highest priority. As there are a finite number of students and schools, there must exist a cycle. In each cycle, assign the student to the school she is pointing to. Remove the assigned students and reduce the capacity of each school that is assigned a student by one. When a school has no remaining capacity, it is removed. The algorithm terminates when there are no students remaining or when there are no schools with available capacity. TTC has many desirable features. It is Pareto efficient, strategyproof, and individually rational. For the housing market setting introduced by Shapley and Scarf (1974), TTC makes the unique assignment that is in the core (Roth and Postlewaite, 1977).

In TTC, if a student i's most preferred school is a and the student has one of the q_a highest priorities at a, then she is always assigned a. However, TTC allows her to trade her priority at other schools in order to be assigned a. This causes an unnecessary distortion as the following example illustrates.

Example 1 Suppose there are three agents $\{i,j,k\}$ and two schools $\{a,b\}$. School a has a capacity of two while school b has a capacity of one. Define student preferences,

[1] When objects do not have priorities over agents, then ex-ante fairness is typically interpreted as equal treatment of equals. See for example Hylland and Zeckhauser (1979), Bogomolnaia and Moulin (2001, 2002), and Klaus and Klijn (2006). Kesten and Yazici (2012) consider ex-post fair assignment rules when objects do not have priorities.

[2] As there is no tension between efficiency and strategyproofness, this is equivalent to valuing efficiency first, strategyproofness second, and fairness third.

P, and school priorities, $\succ$, according to the following rank-order lists:

P_i	P_j	P_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 is fair and efficient whereas the TTC assignment is not fair. The only agent that does not get her top choice is i, and i has lower priority at her top choice, b, then the student assigned to b, k.

We introduce two variations of TTC in order to avoid these unnecessary trades. In the first algorithm, Clinch and Trade (C&T), we run TTC but at each step of the algorithm we check if an agent is able to "clinch" her most preferred school before we have her point to it. An agent clinches a if a is her most preferred school and she has one of the q_a highest priorities at a, where q_a is the remaining capacity of a. This clinching process allows us to reduce the number of unnecessary trades implemented by TTC. Note that it does not completely eliminate these trades as once an agent points at a school a, she continues to point at a until a is assigned to capacity. This is necessary in order to preserve strategyproofness,[3] but unfortunately, it allows her to trade for a even if in subsequent rounds she has one of the q_a highest priorities at a.

C&T differs from TTC in several ways. We will demonstrate that unlike TTC, C&T is not independent of the order in which cycles are processed. In addition, C&T is bossy and therefore not group-strategyproof.

We introduce a second algorithm, First Clinch and Trade (FC&T), which is strategyproof, efficient, nonbossy, group strategyproof, and independent of the order in which cycles are processed. We define an agent i to be guaranteed a seat at school a if i has one of the q_a highest priorities at school a. In this algorithm, we run TTC but if an agent ever points at a school where she was initially guaranteed a seat, we assign the student to the school without allowing her to trade her priority.

FC&T has implications for two important characterizations of assignment mechanisms. Papai (2000) demonstrates that a mechanism is group-strategyproof, Pareto efficient, and reallocation proof if and only if it is a hierarchical exchange rule. In a recent paper, Pycia and Unver (2010) generalize hierarchical exchange rules to a new class of assignment mechanisms that they call Trading Cycles. They demonstrate that Trading Cycles is the unique group-strategyproof and Pareto efficient class of mechanisms.

[3] See Example 4 on Page 8.

The characterizations of hierarchical exchange rules and Trading Cycles are both for the special case where objects may only be assigned to one agent. A natural question is whether or not they generalize when capacities may be greater than one. The simplest way to generalize hierarchical exchanges to general capacities is to allow only one agent to own an object at any given time. This is the approach taken by Pycia and Unver (2011). We call this a *single hierarchical exchange*. In Section 4 we demonstrate that FC&T is not a single hierarchical exchange or an instance of trading cycles for any simple ownership. Since FC&T satisfies all of the axioms of Papai (2000) and Pycia and Unver (2010), this demonstrates that a more nuanced ownership structure must be considered to fully characterize exchange mechanisms for school choice. Dur (2013) and Morrill (2013) provide characterizations of TTC for the school choice problem.

The paper most similar to ours 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). 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. ETTC is strategyproof and Pareto efficient.

ETTC and our algorithms were developed independently and consequently take different approaches; however, ETTC addresses the same issue with TTC that is described in the current paper. 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. Example 2 demonstrates this.

Example 2 Suppose there are four agents $\{i,j,k,l\}$ and three schools $\{a,b,c\}$. School a has a capacity of two while school b and c have a capacity of one. Define student preferences, P, and school priorities, $\succ$, according to the following rank-order lists:

P_i	P_j	P_k	P_l	$\succ_a$	$\succ_b$	$\succ_c$
b	a	b	c	l	j	l
a	b	a		i	k	
				j	i	
				k	l	

In C&T, l clinches c. After removing l, j is able to clinch a. This enables k to clinch b, and finally, i is assigned to a.

In ETTC, the initial student-school pairs are $\{(l,a),(i,a),(j,b),(l,c)\}$. (l,c) points to itself. (l,a) points to (l,c). (i,a) points to (j,b). (j,b) points to (i,a) since i is ranked higher at b than l is. Therefore, in the first round of ETTC, i, j, and l are assigned to b, a, and c, respectively. Therefore, k is assigned to a.

Note that FC&T does not iterate the clinching procedure. In Example 2, FC&T makes the same assignment as ETTC. However, ETTC and the clinching algorithms also differ in the manner in which they make trades. The types of trades made in C&T and FC&T correspond exactly to trading cycles. However, the trades in ETTC are more complex. Example 3 illustrates this.

Example 3 Suppose there are four agents $\{i,j,k,l\}$ and three schools $\{a,b,c\}$. School a has a capacity of two while b and c have a capacity of one. Define student preferences, P, and school priorities, $\succ$, according to the following rank-order lists:

P_i	P_j	P_k	P_l	$\succ_a$	$\succ_b$	$\succ_c$
a	c	b	b	j	i	i
				l	j	l
				j	k	j
					l	

In C&T (and FC&T) no agent is able to clinch in the first round. $\{i,a,j,c\}$ is the only cycle. In the second round of C&T, k clinches b. In the second round of FC&T, k and b form a trivial cycle. Therefore, both algorithms assign i, j, k, and l to a, c, b, and a, respectively.

In ETTC, the initial student-school pairs are $\{(j,a),(l,a),(i,b),(i,c)\}$. (j,a) points to (i,c) since j's favorite school is c. (i,c) points to (l,a) since i's favorite school is a but l is ranked higher than j at c. (l,a) points to (i,b), and (i,b) points to (j,a). Therefore, ETTC assigns i, j, k, and l to a, c, a, and b, respectively.

1 Model

We consider a finite set of agents $I=\{i,j,k,\ldots\}$ and a finite set of objects $O=\{a,b,c,\ldots\}$. We assume that each object a has a capacity for q_a many agents. Each agent $i\in I$ has a complete, irreflexive, and transitive preference relation P_i over $O\cup\{\emptyset\}$. $\emptyset$ represents an agent being unassigned, and $q_\emptyset=\infty$. aP_ib indicates that i strictly prefers object 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 object $a \in O$ has a complete, irreflexive, and transitive priority ranking $\succ_a$ over I. In particular, $i \succ_a j$ is interpreted as agent i has a higher priority for object a than agent j.

We let $P=(P_i)_{i\in I}$, $\succ = (\succ_a)_{a\in O}$, $P_{-I'}=(P_i)_{i\in I\setminus I'}$, and $\succ_{-O'} = (\succ_a)_{a\in O\setminus O'}$. When I and O are clear from the context, for notational convenience we will refer to the assignment problem as $(P,\succ)$.

An **assignment** is a function $\mu : I \to O \cup \{\emptyset\}$ such that for each $a \in O$, $|\{i \in I | \mu(i)=a\}| \leq q_a$. In a slight abuse of notation, for a set of agents $I' \subset I$, we define $\mu(I') = \{a \in O | \exists i \in I'$ such that $\mu(i)=a\}$, and set $\mu(a)=\{i \in I|\mu(i)=a\}$.

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.

We denote by $\mathscr{R}$, $\mathscr{C}$, and $\mathscr{A}$ the sets of all possible preference relationships, priority rankings, and assignments, respectively. An **assignment mechanism** is a function $\phi : \mathscr{R} \times \mathscr{C} \to \mathscr{A}$. The following are important and well studied properties of assignment mechanisms.

- A mechanism ϕ is **strategyproof** if reporting true preferences is each agent's dominant strategy. That is:

$$\phi(P,\succ)(i)R_i\phi(P_i',P_{-i},\succ)(i)$$

  for all P, $\succ$, $i \in I$, and P_i'.[4]
- A mechanism ϕ is **nonbossy** if for all P, $\succ$, $i \in I$, and P_i', $\phi(P,\succ)(i)=\phi(P_i',P_{-i},\succ)(i)$ implies $\phi(P,\succ)=\phi(P_i',P_{-i},\succ)$.
- A mechanism ϕ is **group-strategyproof** if for all P and $\succ$, there does not exist $J \subset I$ and P_J' such that for all $i \in J$, $\phi(P_J',P_{-J},\succ)(i)R_i\phi(P,\succ)(i)$ and for some $j \in J$, $\phi(P_J',P_{-J},\succ)(j)P_j\phi(P,\succ)(j)$.
- A mechanism ϕ is **manipulable through reallocation** if there exist P, $i,j \in I$, and P_i',P_j' such that $\phi(P_i',P_j',P_{-i,j},\succ)(i)R_j\phi(P,\succ)(j)$, $\phi(P_i',P_j',P_{-i,j},\succ)(j)P_i\phi(P,\succ)(i)$ and $\phi(P,\succ)(h)=\phi(P_h',,P_{-h},\succ)(i) \neq \phi(P_i',P_j',P_{-i,j},\succ)(h)$ for $h=i,j$. A mechanism is **reallocation-proof** if it is not manipulable through reallocation.

Abdulkadiroglu and Sönmez (2003) give a detailed description 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 $\{o_1,i_1,\ldots,o_K,i_K\}$ such that each o_j and 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. For any $R \in \mathscr{R}$, $\succ \in \mathscr{C}$, the mechanism $TTC(R,\succ)$ outputs the assignment made by TTC.

[4] Note that we only consider the strategic incentives of students. Priorities at a school are typically created by a school board and therefore are not prone to manipulation. However, see Kesten (2012) and Afacan (2014) for interesting potential manipulations by schools.

2 Clinch and Trade

In this section, we introduce a new algorithm, Clinch and Trade (C&T), designed to reduce the number of unnecessary trades implemented by TTC. C&T is a variation of TTC. Each student points to her favorite school, and each school points to the student with highest priority. The innovation is to check if an agent is guaranteed admissions to her favorite school before allowing her to point at it. TTC is independent of the order in which we process cycles; however, the order in which we process cycles is important to C&T. In particular, when an agent is assigned to a school and removed, she is removed from the priority list of each school. As a result, the priorities of each remaining student weakly improve. Therefore, a student may not initially be guaranteed admissions to a school but after other students are assigned and removed, her admission may become guaranteed.

However, we must be careful about which students we allow to clinch. In particular, once a student has pointed at school a, we cannot let her clinch a. This is necessary to preserve strategyproofness. All students participate in the clinching process in the first round, but we only allow a student to participate in the clinching process in round k if the school she was pointing to in round $k-1$ was removed at the end of round $k-1$.

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 $q_a = q_a - 1$. Whenever we remove a student, we adjust the rankings of all schools accordingly. We call this *clinching* a school. Iterate the clinching procedure until no agent has one of the q_a highest rankings 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 agent to the school she is pointing to, remove the agent, and reduce the capacity of the school by one.

Round k:

k.a If the school that i was pointing to in Round $k-1$ still has available capacity, then i continues to point to the same school. For the students whose favorite school was removed in the previous round, iterate the clinching process until no student has one of the q_a highest priorities at her most preferred school a unless she was pointing to a at the end of round $k-1$.

k.b 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 agent to the school she is pointing to, remove the agent, and reduce the capacity of the school by one.

The most natural variation of TTC would be to have an agent i clinch an object a whenever she points to a and has one of the q_a highest priorities at a. We call this variation of Always Clinch and Trade (AC&T). In contrast, C&T only allows an agent to clinch an object before the first time she points at it. Example 4 demonstrates that AC&T is not strategyproof.

Example 4 Suppose there are five agents $\{i,j,k,l,m\}$ and four schools $\{a,b,c,d\}$. School a has a capacity of two while the other schools have a capacity of one. Define student preferences, P, and school priorities, $\succ$, according to the following rank-order lists:

P_i	P_j	P_k	P_l	P_m	$\succ_a$	$\succ_b$	$\succ_c$	$\succ_d$
c	a	b	d	c	i	j	l	m
b					l	k		
a					j	i		
					k			

In the first round of AC&T, no agent is able to clinch an object. The only cycle is $\{l,d,m,c\}$. After removing this cycle, j is now able to clinch a. k now clinches b, and i is assigned to a. Therefore, AC&T assigns i, j, k, l, and m to a, a, b, d, and c respectively. Suppose alternatively that i ranks b first among schools. Now $\{i,b,j,a\}$ form a cycle in the first round of AC&T and i is assigned to b, a school she strictly prefers to the assignment she receives under her true preferences. Therefore, AC&T is not strategyproof. Note that the first round of C&T is the same as the first round of AC&T. However, in the second round j does not clinch a. Even though she now has one of the two highest priorities at a, j is not eligible to clinch a in the second round because she was already pointing at a in the first round. As a result, i has no incentive to misrepresent her preferences as she is assigned b under C&T.

Although AC&T is not strategyproof, the following proposition demonstrates that C&T is strategyproof and efficient.

Proposition 1 *C&T is:*

- *Pareto efficient*
- *strategyproof*

Proof *Pareto efficient:* A student assigned in Round 1a cannot be made better off. Consider an agent i assigned in Round 1b. If i receives her top choice, then she cannot be made better off. If not, then i's top choice was assigned up to capacity to agents in Round 1a. Therefore, i cannot be made better off without making another agent worse off. Iterating this argument, we find that no student can be made strictly better off without harming a student who was assigned earlier in the algorithm.

Strategyproof: Consider the following thought exercise. We fix a student i, make i aware of the preferences of all other students, and at the beginning of each round, we give i the opportunity to change her preferences if she desires. We show that she never needs to change her preferences in the current round, and therefore never needs to change her preferences at all. i never needs to change her preferences before an "a" round because if she clinches an object, it is her top choice among remaining objects. If she changes her preferences but does not clinch, then she does just as well waiting until the "b" round to change her preferences. Consider i's choice before a "b" round. If she will be assigned in this round with her true preferences, then she does not benefit from changing her preferences as she already receives her top choice among available objects. Suppose she is not going to be assigned in that round. If she changes her preferences and this does not cause a cycle to form, then she does just as well leaving her preferences the same as she can always change them before the next round. Suppose changing her preferences causes the cycle $\{i, o_1, i_1, o_2, \ldots, i_{k-1}, o_k\}$ to form. In this round, i is the only agent that can form a cycle with any of the agents $\{i_1, \ldots, i_{k-1}\}$. Therefore, if i does not change her preferences, then $\{i_1, \ldots, i_{k-1}\}$ continue to point to the same objects in the next round. In particular, consider any $i_j \in \{i_1, \ldots, i_{k-1}\}$. i_j does not clinch an object in the next round because she is pointing to an object that was not removed in the previous round. Since i_j has the highest priority at o_j, o_j cannot be assigned to capacity until i_j is assigned. Similarly, i_j continues to point to o_{j+1} until o_{j+1} is removed. Therefore, if i does not change her preferences, then o_j continues to point to i_j in the next round and i_j continues to point to o_{j+1} for each j. Therefore, i does not need to change her preferences in this round as she can change her preferences in the next round and still be assigned o_1. Therefore, there is never a round where i needs to change her preferences to improve her assignment. Since i never needs to change her preferences in the current round but she is assigned in a finite number of rounds, i never needs to change her preferences and the mechanism is strategyproof.

While C&T is strategyproof and efficient, it no longer retains some of TTC's desirable properties. We demonstrate this through a series of examples. First, C&T is bossy (and therefore is not group strategyproof, Papai 2000). The reason for this is that although C&T reduces the number of unnecessary cycles, it does not completely eliminate them. Next, a desirable feature of TTC is that it makes the same assignment regardless of the order in which cycles are chosen. C&T is dependent of the order in which cycles are chosen.

Example 5 (C&T is bossy) Suppose there are five agents $\{i, j, k, l, m\}$ and four schools $\{a, b, c, d\}$. The capacities are $q_a = 2$ and $q_b = q_c = q_d = 1$. Define P and $\succ$ according to the following rank-order lists:

P_i	P_j	P_k	P_l	P_m	$\succ_a$	$\succ_b$	$\succ_c$	$\succ_d$
b	a	b	c	d	i	j	m	l
a	b	a			l	k		
c	c	c			j	i		

No agent clinches a school. In Round 1b $\{l, c, m, d\}$ and $\{i, b, j, a\}$ form cycles. Therefore, the final assignment is:

$$\begin{pmatrix} i & j & k & l & m \\ b & a & a & c & d \end{pmatrix}$$

However, if j submits preferences $P'_j : c, a, b$, then she points to c in Round 1b. Since c is assigned to l and removed during 1b, in Round 2a j clinches a (l was removed in 1b, therefore j now has one of the two highest priorities at a school with capacity two). Now k clinches b and i clinches a. So the final assignment is:

$$\begin{pmatrix} i & j & k & l & m \\ a & a & b & c & d \end{pmatrix}$$

Therefore, j is bossy as she can change her preferences, not change her assignment, but change the assignment of other agents.

The next example demonstrates that C&T is dependent on the order in which cycles are removed.

Example 6 Suppose all schools have a capacity for one student except s_1 which has a capacity of two. Consider the following preferences and priorities:

i_1	i_2	i_3	i_4	i_5	i_6	i_7	s_1	s_2	s_3	s_4	s_5	s_6
s_3	s_2	s_2	s_4	s_3	s_6	s_5	i_7	i_1	i_4	i_5	i_6	i_7
s_1	s_1	s_1					i_3	i_2				
							i_1	i_3				

No student clinches a school in Round 1.a. However, there are two cycles: (i_4,s_4,i_5,s_3) and (i_6,s_6,i_7,s_5). If we only assign cycle (i_4,s_4,i_5,s_3) then i_1 does not clinch s_1. Therefore, she points to s_1. The only cycle is (i_6,s_6,i_7,s_5). However, after we remove the cycle, i_1 continues to point to s_1 (only a student who's most preferred assignment was removed in the previous round may clinch a school). Therefore, the reduced problem is:

i_1	i_2	i_3	s_1	s_2
s_1	s_2	s_2	i_3	i_1
	s_1	s_1	i_1	i_2
				i_3

(i_1,s_1,i_3,s_2) forms a cycle, and the final assignment is:

$$\begin{pmatrix} i_1 & i_2 & i_3 & i_4 & i_5 & i_6 & i_7 \\ s_1 & s_1 & s_2 & s_4 & s_3 & s_6 & s_5 \end{pmatrix}$$

However, if in Round 1.b. we process the cycle (i_6,s_6,i_7,s_5) instead of the cycle (i_4,s_4,i_5,s_3), then every student continues to point to the same school. Therefore, we next process the cycle (i_4,s_4,i_5,s_3). Now, i_1 is no longer pointing to a school and the reduced problem is:

i_1	i_2	i_3	s_1	s_2
s_1	s_2	s_2	i_3	i_1
	s_1	s_1	i_1	i_2
				i_3

Therefore, i_1 clinches s_1. After we remove i_1, i_2 forms a cycle with s_2 and the final assignment is:

$$\begin{pmatrix} i_1 & i_2 & i_3 & i_4 & i_5 & i_6 & i_7 \\ s_1 & s_2 & s_1 & s_4 & s_3 & s_6 & s_5 \end{pmatrix}$$

3 First Clinch and Trade

We introduce an alternative modification of TTC that is nonbossy and independent of the order in which cycles are processed. Intuitively, we run TTC, but we do not allow a student that initially had one of the q_a highest priorities at a school a to trade with another student in order to receive a. The key difference between First Clinch and Trade (FC&T) and C&T is that in FC&T we do not update who is able to clinch her most preferred school. An agent is only able to clinch school a if she initially had one of the q_a highest priorities at a.

For notational convenience, we fix the priorities of the schools, $\succ$. We define a student's **ranking** at school a to be her spot on the priority list, $r_a(i) := |\{j \in I | j \succeq_a i\}|$. For each school a, we define the set of students that are *guaranteed* admissions to a:

$$G_a := \{i \in I | r_a(i) \leq q_a\}.$$

In particular, a student is only defined to be guaranteed admissions to a school a if she initially has one of the q_a highest rankings at a.

First Clinch and Trade (FC&T)[5]

To aid with intuition and the proofs of properties of the algorithm, we provide two equivalent formulations.

Formulation 1 - "Pointing" version

Round k:

Each student points to her most preferred school with available capacity.

Each school points to the remaining student with highest priority.

If student i is pointing at school a and $i \in G_a$ then assign i to a, remove i, and reduce a's capacity by one. For the remaining students, if there exists a cycle, assign each student in the cycle to the school she is pointing to. Remove the students in the cycle and reduce the capacity of each school in the cycle by one.

The algorithm terminates when all students are assigned or no school has available capacity.

We will say that a student is assigned *directly* to a if $i \in G_a$. A student is assigned *indirectly* to a if it is part of a non-trivial trading cycle. Note that unlike TTC, two different students may be assigned to the same school in the same round in FC&T. This occurs if a is both involved in a trading cycle and there is a student i such that i points to a, i is not the highest priority student for a, but $i \in G_a$. Note however that no school is assigned more than its capacity by the algorithm.

The pointing formulation is described as processing the direct assignments first and the indirect assignments second. However, the algorithm is well defined for any order in which assignments are made. In fact, we will demonstrate that like TTC, the final assignment is independent of the order in which assignments are processed.

Formulation 2 - "Clone" version

[5] In early drafts of this paper, we called this algorithm Priority-Adjusted TTC.

For our second formulation, we use a cloning procedure that is similar but not equivalent to the cloning procedure described in Roth and Sotomayor (1990) for generalizing the marriage problem to the college admissions problem when schools have substitutable preferences. Consider any assignment problem I, O, R, $\succ$. As in the standard cloning procedure, we clone each school $a \in O$ q_a times. Formally, we define schools $a^{(1)}, a^{(2)}, \ldots, a^{(q_a)}$. For expositional purposes, we call $a^{(i)}$ the i^{th} *clone* of a, and we call a the *prototype* of $a^{(i)}$. Each clone has the capacity for one student, and a clone has the same priorities as the prototype with one exception: the k^{th} ranked student is moved to the top of the priority list for the k^{th} clone. Student preferences over clones are modified in a similar manner. If a student prefers a to b, then that student prefers any of the a clones to any of the b clones. A student strictly prefers a lower numbered clone to a higher numbered clone with one exception: if student i was initially the k^{th} ranked student at a where $k \le q_a$ (in other words, if i was initially guaranteed a seat at a), then i's favorite a-clone is $a^{(k)}$. We designate this induced assignment problem by $\hat{I}$, $\hat{O}$, $\widehat{R}$, $\hat{\succ}$.

We define the clone version of the algorithm, $\chi(R, \succ)$, by assigning each agent i to the prototype of $TTC(\widehat{R}, \hat{I}, \hat{O}, \hat{\succ})(i)$. In other words, if $TTC(\widehat{R}, \hat{I}, \hat{O}, \hat{\succ})$ assigns i to a clone of school a, then $\chi(R, \succ)$ assigns i to a. For notational convenience, when the agents, objects, and priorities are clear from context, we will use the notation $TTC(\widehat{R})$ and $\chi(R)$.

The pointing and the clone formulations are equivalent. Specifically, for any order in which agents are processed in the Pointing version, there is a corresponding order of processing agents in the Clone version that yields the same assignment, and *vice versa*. To see this, consider any order of processing agents under the pointing formulation. At every step, we either process a student who clinches a school or else a group of students that are part of a cycle. Student j only clinches school a in the pointing formulation if j was initially guaranteed a spot at a and a is now j's favorite school with available capacity. In particular, if j is initially the k^{th} ranked student at a, then $k \le q_a$. Therefore, by construction j is the highest ranked student at $a^{(k)}$ in the clone formation and $a^{(k)}$ is now j's favorite object; therefore, j and $a^{(k)}$ form a trivial cycle in the clone formulation. Similarly, in a non-trivial cycle under the pointing formation, no agent was guaranteed the object she is pointing to; however, she is the highest ranked student of the school pointing at her. Therefore, under the clone formulation, the same agents and the lowest numbered remaining clones form a cycle. Analogously, each cycle in the clone formulation corresponds to either a cycle or a clinch in the pointing formulation.

We can immediately conclude that the pointing formulation is independent of the order in which we process the students. Processing students in two different orders in the pointing formulation corresponds exactly to processing cycles in two different orders under the clone formulation. However, the clone formulation simply runs TTC with modified objects and preferences. Since TTC is independent of the order in which cycles are processed, the clone formulation makes the same assignment under either ordering. Therefore, the pointing formulation must make the same assignment under either ordering as well.

Note that FC&T does not process as many students via clinching as C&T. The reason is that a student i may not initially be guaranteed a spot at school a, but if students ranked higher than i are assigned to schools other than a, i may become guaranteed a spot at a in a later round. However, FC&T does not keep i from trading her priority at a different school b in order to receive a spot at a unless i is initially guaranteed a spot at a. Therefore, FC&T still allows unnecessary trades to take place.

However, the disadvantage of C&T is that it lost several of the desirable attributes of TTC. Specifically, C&T is bossy and dependent on the order in which cycles are processed. As the next proposition demonstrates, FC&T retains the desirable properties of TTC that have been previously identified by the literature. Note that although FC&T is closely related to TTC, it is not equivalent to TTC for any priority structure. We demonstrate this in Section 4.

Proposition 2 *First Clinch and Trade is:*

1. *Pareto efficient.*
2. *strategyproof.*
3. *non-bossy.*
4. *group strategyproof.*
5. *reallocation proof.*
6. *independent of the order in which cycles are processed.*

We will use the clone formulation in the proof of Proposition 2. The intuition is that since FC&T is an instance of TTC under a modified problem, the algorithm has many of the same properties as TTC. The following technical lemma will be useful in our argument. Suppose i has two preference profiles, R_i and R_i', such that yield the same assignment for i under FC&T: $\chi(R)(i) = \chi(R_i', R_{-i})(i) = a$. Potentially, i could have been assigned to two different clones of a. We establish that this is not the case. If i has the same assignment under two different preference profiles, then i was assigned to exactly the same clone under either preference.

Lemma 1 *For any $i \in I$, $R \in \mathscr{R}$ and any R_i', $\chi(R)(i) = \chi(R_i', R_{-i})(i)$ if and only if $TTC(\widehat{R})(i) = TTC(\widehat{R_i'}, \widehat{R_{-i}})(i)$.*

Proof The if direction is trivial. By the definition of FC&T, if $TTC(\widehat{R})(i) = TTC(\widehat{R_i'}, \widehat{R_{-i}})(i)$, then $\chi(R)(i) = \chi(R_i', R_{-i})(i)$. For the other direction, consider any i, R and R_i' such that $\chi(R)(i) = \chi(R_i', R_{-i})(i)$. We verify that i is assigned to the same clone under both R_i and R_i'. When running TTC, do not process a cycle involving i until all other cycles have been processed. In the round where i is processed, note that there is a path from all remaining schools to i as otherwise there would be a cycle not containing i. Also, note that the structure of this graph does not depend on i's report. In particular, it is the same whether i reports R_i or R_i'. Let $a = \chi(R)(i) = \chi(R_i', R_{-i})(i)$. If $i \in G_a$ then i points to $a^{(r_a(i))}$ under either R_i or R_i'. Otherwise, i points to the lowest numbered clone that remains. But since the remaining schools are the same, and since under both R_i and R_i' i points to a clone of a, it must be that i points to the same clone under either R_i or R_i'. Therefore, $TTC(\widehat{R})(i) = TTC(\widehat{R_i'}, \widehat{R_{-i}})(i)$.

Proof Fix the set of students, schools, priorities, and capacities, and let $\chi(R)$ be the assignment produced by FC&T for the set of preferences R.

Pareto efficiency - Suppose for contradiction there exists a Pareto improvement which changes the assignment of agents $A = \{a_1, a_2, \ldots, a_n\}$. Let a_i be the agent in A that was assigned in the earliest round of the pointing formulation. a_i already receives her most preferred remaining school. Therefore, if she is reassigned, it must change the assignment of an agent assigned in an earlier round. This contradicts a_i being the earliest reassigned agent.

Strategyproofness - Consider any agent $i \in I$. In the clone formulation, process the cycle containing i last. As we mentioned previously, i's report does not affect the set of schools that remain, and there must exist a path from every remaining school to i. Therefore, i has her choice among remaining schools and cannot do better than revealing her preferences truthfully and being assigned her most preferred remaining school.

Nonbossy This follows from Lemma 1 and the fact that TTC is nonbossy. Consider any R and R'_i such that $\chi(R)(i) = \chi(R'_i, R_{-i})(i)$. Let $R' = \{R'_i, R_{-i}\}$. By Lemma 1, $TTC(\hat{R})(i) = \chi(R)(i) = \chi(R'_i, R_{-i})(i) = TTC(\hat{R}')(i)$. Since TTC is non-bossy, $TTC(\hat{R})(j) = TTC(\hat{R}')(j)$ for every $j \neq i$. Therefore, $\chi(R)(j) = TTC(\hat{R})(j) = TTC(\hat{R}')(j) = \chi(R'_i, R_{-i})(j)$ for every $j \neq i$.

Group-Strategyproof Papai (2000) demonstrates that a mechanism is group-strategyproof if and only if it is strategyproof and nonbossy.

Reallocation-proof This is similar to the argument for nonbossiness. Suppose for contradiction there exist P, i, j, P'_i, and P'_j such that

$$\chi(P'_i, P'_j, P_{-\{i,j\}})(i) \; R_j \; \chi(P)(j) \tag{1}$$

$$\chi(P'_i, P'_j, P_{-\{i,j\}})(j) \; P_i \; \chi(P)(i) \tag{2}$$

$$\chi(P)(h) = \chi(P'_h, P_{-h})(i) \neq \chi(P'_i, P'_j, P_{-\{i,j\}})(h) \text{ for } h = i, j. \tag{3}$$

Since by Lemma 1 $\chi(Q) = TTC(\hat{Q})$ for all Q, it must be that

$$TTC(\hat{P}'_i, \hat{P}'_j, \hat{P}_{-\{i,j\}})(i) \; \hat{R}_j \; TTC(\hat{P})(j) \tag{4}$$

$$TTC(\hat{P}'_i, \hat{P}'_j, \hat{P}_{-\{i,j\}})(j) \; \hat{P}_i \; TTC(\hat{P})(i) \tag{5}$$

$$TTC(\hat{P})(h) = TTC(\hat{P}'_h, \hat{P_{-h}})(i) \neq TTC(\hat{P}'_i, \hat{P}'_j, \hat{P}_{-\{i,j\}})(h) \tag{6}$$

for $h = i, j$. But this violates the fact that TTC is reallocation-proof.

4 Relationship to Hierarchical Exchange Rules

Papai (2000) introduces an important class of assignment mechanisms: hierarchical exchange rules. Hierarchical exchange rules are a generalization of TTC. Initially, ownership rights to each object are assigned to some agent. Agents then trade as in TTC. If an agent "owns" multiple objects, then when that agent is assigned, the objects that she did not use in her trading cycle are inherited by remaining agents. The hierarchical exchange rule is determined by the initial endowments and the inheritance rule. For example, a serial dictatorship is a hierarchical exchange rule where the dictator is initially endowed all objects and the next dictator inherits all remaining objects.

Hierarchical exchanges are a broad and complex class of mechanisms, but the central idea is rather intuitive. If we assign property rights and allow the agents to trade, the resulting assignment will be Pareto efficient. When an object can only be assigned to one agent, then the ownership structure is straightforward; each object is owned by exactly one agent. However, when objects may be assigned to multiple agents, the ownership structure may be more complex.

The simplest ownership structure is to allow each object a be owned by only one agent i. After i is assigned, if a still has available capacity, then a is inherited by one of the remaining agents. This is the approach taken in Pycia and Unver (2011) and we will refer to this as a **single hierarchical exchange**. Not only is this a natural ownership rule, but it means there is only one type of trade that we need to consider: a trading cycle.

When objects may be assigned to only one agent, Papai (2000) demonstrates that a mechanism is group-strategyproof, Pareto-optimal, and reallocation-proof if and only if it is equivalent to a hierarchical exchange rule. For general capacities, we demonstrate that these axioms are not sufficient to categorize the class of single hierarchical exchanges. We do not need to worry about precise definitions of inheritance rules or order of exchange because our counterexample is based on property rights. The essential point is that the clinching procedure cannot be captured by a simple ownership structure. Consider an agent who is guaranteed an object a but does not have the highest priority at a. This agent does not own a since she is unable to trade a. However, she may be assigned a without trading for it. In a single hierarchical exchange rule, if you can be assigned an object without trading for it, you must own it. If you own it, then you are able to trade it for other objects.

In particular, FC&T is more nuanced than simply assigning property rights and allowing agents to trade. We show that irrespective of the inheritance procedure, there is no initial allocation of property rights to agents so that FC&T corresponds to a single hierarchical exchange rule.

Proposition 3 *First Clinch and Trade is not equivalent to any single hierarchical exchange rule.*

Proof Suppose there are three agents $\{i,j,k\}$ and three objects $\{a,b,c\}$. Object a has a capacity of two while objects b and c have capacity one. Define $\succ$ according to the following rank-order lists:

$\succ_a$	$\succ_b$	$\succ_c$
i	j	k
j	k	i
k	i	j

We fix $\succ$ and let $\mu(R)$ be the assignment made by FC&T for a given R. Suppose for contradiction that there exists some single hierarchical exchange rule λ that corresponds to μ. That is to say, for given preferences R, $\mu(R) = \lambda(R)$. We define the following preferences for convenience.

P_i	P_j	P_k	P'_i	P'_j	P'_k	P''_i	P''_j	P''_k
b	a	b	c	c	a	a	a	a
a	b	a	a	a	b	b	b	b
c	c	c	b	b	c	c	c	c

It is straightforward to verify that under λ, b is initially owned by j and c is initially owned by k. Since λ is a hierarchical exchange rule, initially some agent must own a. Suppose this agent is i. Then $\lambda(R)(i) = b$ since i trades a to agent j in order to receive b. But this is a contradiction as $\mu(R)(i) = a$. Therefore, a is not owned by i. Suppose instead that a is owned by j. Then $\lambda(R')(j) = c$ as j trades a for c with agent k. However, $\mu(R')(j) = a$,[6], a contradiction. Therefore, $\lambda(R')(j) \neq \mu(R')(j)$ a contradiction. Therefore, a is not owned by j. Finally, suppose that a is owned by k. Then $\lambda(R'')(k) = a$, a contradiction since $\mu(R'')(k) = b$. Therefore, no agent owns a which is itself a contradiction.

Our algorithm also relates to a second important characterization. Pycia and Unver (2010) introduce a generalization of hierarchical exchange rules called Trading Cycles. Trading Cycles extends the ownership structure of hierarchical exchange rules to allow for two types of control over an object: ownership and brokerage. There can be at most one broker and at most one brokered house. Each house points to the agent that controls it. Each agent except for the broker points to her favorite object. The broker, if there is one, points to her favorite object that is not the brokered house. Otherwise, Trading Cycles proceeds in an identical manner to a hierarchical exchange rule.

Pycia and Unver (2010) demonstrates when objects have capacity for at most one agent that any group-strategyproof and Pareto efficient assignment mechanism is equivalent to an instance of Trading Cycles. It is straightforward to generalize the argument in Proposition 3 to show that FC&T is not an instance of Trading Cycles if there is a simple ownership structure. In particular, none of the agents act as a broker as all agents may be assigned to any object. FC&T does not correspond to Trading Cycles for the same reason it does not correspond to a single hierarchical exchange rule; FC&T has a more nuanced ownership structure.

In addition to single hierarchical exchanges, there is at least one other natural ownership structure when objects may be assigned to multiple agents. Effectively, in a single hierarchical exchange, one student owns all q_a seats at a school a. An alternative is to allow an object a with capacity q_a to be owned by up to q_a different agents. We call this **multiple hierarchical exchange**. One advantage to single hierarchical exchanges is that there is only one natural way for the students to trade: a top trading cycle. However, when multiple students own the same school, we must also specify how the agents trade objects. For example, the clone formulation demonstrates that FC&T can be considered a multiple hierarchical exchange. There, we specify which trades are made by specifying the preferences agents have over clones of the same object.

5 Conclusion

Typically a school board chooses between Top Trading Cycles and the Deferred Acceptance algorithm when deciding a mechanism to assign students to schools. However, this paper argues that alternative mechanisms should be considered. In particu-

[6] No agent is guaranteed her top choice and i and k form the first cycle.

lar, a student who is guaranteed to be admitted to her favorite school should not be allowed to trade her priority at other schools. These trades are irrelevant to her and may violate the priorities of the other students.

We propose a simple solution to this problem. If a student is guaranteed admission to her favorite school, then she should be assigned to that school and not allowed to trade. We introduce two strategyproof and efficient mechanisms that mitigate this distortion: Clinch and Trade and First Clinch and Trade.

References

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

Afacan, M. 2014. Fictitious Students Creation Incentives in School Choice Problems. *Economic Theory*.

Balinski, M., and T. Sönmez. 1999. A tale of two mechanisms: Student placement. *Journal of Economic Theory* 84 (1): 73–94.

Bogomolnaia, A., and H. Moulin. 2001. A new solution to the random assignment problem. *Journal of Economic Theory* 100 (2): 295–328.

Bogomolnaia, A., and H. Moulin. 2002. A simple random assignment problem with a unique solution. *Economic Theory* 19 (3): 623–636.

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

Gale, D., and L. S. Shapley. 1962. College admissions and the stability of marriage. *American Mathematical Monthly* 69 (1): 9–15.

Hylland, A., and R. Zeckhauser. 1979. The efficient allocation of individuals to positions. *The Journal of Political Economy*.

Kesten, O. 2004. Student Placement to Public Schools in US: Two New Solutions. *mimeo*.

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

Kesten, O. 2012. On two kinds of manipulation for school choice problems. *Economic Theory* 51 (3): 677–693.

Klaus, B., and F. Klijn. 2006. Procedurally fair and stable matching. *Economic Theory* 27 (2): 431–447.

Morrill, T. 2013a. An Alternative Characterization of Top Trading Cycles. *Economic Theory* 54: 181–197.

Morrill, T. 2013b. Making just school assignments. *mimeo*.

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

Pycia, M., and U. Ünver. 2011. Trading cycles for school choice. *mimeo*.

Pycia, M., and U. Ünver. 2014. Incentive Compatible Allocation and Exchange of Discrete Resources. *mimeo*.

Roth, A. E. 1982. The economics of matching: Stability and incentives. *Mathematics of Operations Research*.

Roth, A. E., and A. Postlewaite. 1977. Weak versus strong domination in a market with indivisible goods. *Journal of Mathematical Economics* 4 (2): 131–137.

Roth, A. E., and M. A. O. Sotomayor. 1992. *Two-sided matching: A study in game-theoretic modeling and analysis*. Cambridge Univ Pr.

Shapley, L., and H. Scarf. 1974. On cores and indivisibility. *Journal of mathematical economics* 1 (1): 23–37.