Available online at www.sciencedirect.com
ScienceDirect
Journal of Economic Theory 132 (2007) 539–547
JOURNAL OF Economic Theory
www.elsevier.com/locate/jet

Notes, Comments, and Letters to the Editor

# The relationship between top trading cycles mechanism and top trading cycles and chains mechanism

Aradhna Krishna*,[1], Yu Wang

*D5212, Ross School of Business, University of Michigan, 701 Tappan St., Ann Arbor, MI 48109, USA*

Received 8 December 2004
Available online 8 September 2005

**Abstract**

In this paper, we show that there is a relationship between two important matching mechanisms: the Top Trading Cycles mechanism (TTC mechanism proposed by Abdulkadiroglu and Sonmez, 1999) and the Top Trading Cycles and Chains mechanism (TTCC mechanism proposed by Roth, Sonmez, and Unver, 2004). Our main result is that when a specific chain selection rule proposed by Roth et al. is used, these two mechanisms are equivalent. While the equivalence is relevant for one specific case of the TTCC mechanism, it is a particularly interesting case since it is the only version identified by Roth et al. to be both Pareto-efficient and strategy-proof.


*JEL classification:* C71; C78; D71; D78

*Keywords:* Mechanism design; Core; Indivisible goods; Matching

## 1. Introduction

Various types of matching mechanisms have been proposed to solve real-life market design problems. In Abdulkadiroglu and Sonmez [1], the *house allocation problem with*

* Corresponding author.
*E-mail addresses:* aradhna@umich.edu (A. Krishna), yuwz@bus.umich.edu (Y. Wang).

[1] Aradhna Krishna is the Isadore and Leon Winkelman Professor of Marketing and Yu Wang is a doctoral student at the Ross School of Business, University of Michigan.

0022-0531/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jet.2005.07.002

*existing tenants* is formulated as a one-shot discrete resource allocation and exchange problem where all the agents obtain their final allocations simultaneously. To efficiently solve this static one-sided matching problem, Abdulkadiroglu and Sonmez modified Gale's Top Trading Cycles (TTC) algorithm which is used to find the unique core allocation in the context of *housing markets* [6,3]. The variant of TTC mechanism in [1] was proved to be individually rational, strategy-proof, and Pareto-efficient. Hereafter we refer to this version of the TTC mechanism as "A&S TTC" mechanism. [2]

The Top Trading Cycles and Chains (TTCC) mechanism was proposed by Roth et al. [4] to solve a dynamic matching problem—the *kidney exchange problem*. In this problem, some agents may not obtain instant allocations, and instead they get into a queue waiting for desirable items that may become available in the future. Roth et al. proposed to model the waiting queue as a "waitlist option". That is, those who get on the queue are seen as obtaining a waitlist option rather than an instant allocation of real items. Thus Roth et al. were able to consider an instance of this dynamic problem and solve that instance with a static mechanism—TTCC. "Chains" are added to the algorithm (hence the second "C" in the TTCC mechanism) and six "chain selection rules" were proposed to be implemented under the TTCC mechanism. Different chain selection rules reflect different policy objectives and generate different theoretical properties for TTCC. [3]

From a modeling perspective, the house allocation and the kidney exchange problems are different from each other, and the mechanisms that were proposed to solve these problems are different. However, there are obvious similarities between A&S TTC and TTCC. In this paper we prove that when TTCC is implemented with a specific chain selection rule—chain selection rule ***e*** in [4], these two are equivalent algorithms. We state the result formally as follows:

**Proposition 1.** *The Top Trading Cycles and Chains mechanism implemented with chain selection rule* ***e*** *in* [4] (*i.e.*, *TTCC mechanism as proposed by Roth et al.* [4] *to solve the* kidney exchange problem), *and the A&S Top Trading Cycles mechanism* (*i.e.*, *TTC mechanism as proposed by Abdulkadiroglu and Sonmez* [1] *to solve the* house allocation problem with existing tenants) *are equivalent mechanisms*; *chain selection rule* ***e*** *specifies that when multiple chains co-exist*, *one chooses the chain starting with the highest priority agent and keeps it in the system.* [4]

To prove this proposition, we first present a slightly modified version of the kidney exchange model, followed by an introduction of the TTCC mechanism. In specific, TTCC is explained with chain selection rule ***e*** in [4] and TTCC is implemented with this rule. Then we introduce a simplified version of the house allocation model in [1] (we call this "A&S

[2] Interested readers are referred to Chen and Sonmez [2] for an experimental study on the performance of the A&S TTC mechanism, and Sonmez and Unver [7] for a proof of equivalence between an extreme case of the A&S TTC and a core-based mechanism.

[3] Wang and Krishna [8] use a variant of the TTCC mechanism to address the timeshare allocation problem.

[4] The meaning will become clear when we formally present the Top Trading Cycles and Chains mechanism in Section 3.

house allocation model" hereafter), arguing that it is isomorphic to the kidney exchange model. Lastly we explain how the mechanism in [1] (i.e., the A&S TTC mechanism) works and show the equivalence between the TTCC mechanism implemented with rule ***e*** and the A&S TTC mechanism. The last section of this paper is a conclusion of our results.

## 2. Kidney exchange model

The kidney exchange model in [4] consists of:

1. a finite set of item–agent pairs [5] $\{(i_1, a_1), \ldots, (i_n, a_n)\}$ {We denote $I_P$ as the set of all items from the item–agent pairs, i.e., $I_P = \{i_1, \ldots, i_n\}$};
2. a finite set of $n$ at-large items $I_A = \{w_1, w_2, \ldots, w_n\}$;
3. for each agent $a_k$ ($k \in \{1, \ldots, n\}$), a strict preference relation $R_k$ over $I_P \cup I_A$, such that for all the agents:
   (i) $w_1 \succ w_2 \succ w_3 \succ \cdots \succ w_n$, and (ii) $\forall x \in I_P, x \succ w_n \Rightarrow x \succ w_j$ ($j \in \{1, 2, \ldots, n-1\}$), and $\forall y \in I_P, y \prec w_1 \Rightarrow y \prec w_l$ ($l \in \{2, 3, \ldots, n\}$).

In words, in this model there are $n$ agents (kidney recipients or patients), each endowed with one indivisible item (his donor's kidney). We made a slight modification to the original model by explicitly introducing $n$ at-large items (i.e., items that are not paired up with agents at the beginning—$\{w_1, w_2, \ldots, w_n\}$), representing the waitlist options that are available to the patients. These can be interpreted as the first waitlist option to be allocated by the procedure, the second to be allocated, etc. Please note that the number of waitlist options is equal to the number of patients so that all of them can get into the waiting queue if they choose to do so. Each patient $a_k$ has a strict preference relation $R_k$ over all the items. We assume that (i) all agents strictly prefer $w_1$ to $w_2$, $w_2$ to $w_3$, …, $w_{n-1}$ to $w_n$, and (ii) for all agents these at-large items are ranked as a block one after another, so that TTCC can be implemented smoothly using this model. [6] Assumption (i) makes sure that $w_1$ is always allocated before $w_2$, $w_2$ before $w_3$, etc., because for any agent a lower indexed waitlist option is always preferred to a higher indexed waitlist option. Assumption (ii) is necessary because all these waitlist options actually represent the same thing—getting on the waiting queue to wait for an ideal item that shows up in the future. Conceptually there should not exist any item $z \in I_P$ such that $w_{l-1} \prec z \prec w_l$ ($l \in \{2, 3, \ldots, n\}$), i.e., no one will strictly prefer item $z$ to the waitlist option and strictly prefer waitlist option to item $z$ at the same time. This whole setting is equivalent to the original kidney exchange model where there is only one waitlist option $w$, which is allowed to be assigned to multiple agents.

The outcome is a matching of items to agents such that

1. each agent is assigned one item in $I_P \cup I_A$, and
2. each item can only be assigned to one agent.

---

[5] To make the isomorphism between the two models easier to identify, we replace a few terms in the original kidney exchange model with more general terms. Thus, the term "agent" represents "kidney recipient" or "patient", and the term "item" represents "donor's kidney".

[6] We thank an anonymous reviewer for offering this observation.

## 3. TTCC mechanism

TTCC is a procedure consisting of multiple rounds. Items are matched to agents based on identified "cycles" and "chains" [4]. In each round:

- each agent points at an item in $I_P \cup I_A$, and
- each item in $I_P$ points at the agent that it belongs to.

Given the pointing of the agents and the items, a *cycle* is defined as an ordered list of item–agent pairs $(i'_1, a'_1, i'_2, a'_2, \ldots, i'_j, a'_j)$ such that:

- item $i'_1$ points at agent $a'_1$,
- agent $a'_1$ points at item $i'_2$,
- $\vdots$ $\vdots$
- item $i'_j$ points at agent $a'_j$, and
- agent $a'_j$ points at item $i'_1$.

Since each item in $I_P$ points at a unique agent and each agent points at a unique item, if two or more than two cycles coexist, no two cycles can ever intersect, implying that if a cycle is removed from the system, other coexisting cycles are not affected at all.

A *chain* is defined as an ordered list of item–agent pairs $(i'_1, a'_1, i'_2, a'_2, \ldots, i'_j, a'_j)$ such that:

- item $i'_1$ points at agent $a'_1$,
- agent $a'_1$ points at item $i'_2$,
- $\vdots$ $\vdots$
- item $i'_j$ points at agent $a'_j$, and
- agent $a'_j$ points at $w_l$ $(l \in \{1, 2, \ldots, n\})$.

With any chain $(i'_1, a'_1, i'_2, a'_2, \ldots, i'_j, a'_j)$, we will refer to the pair $(i'_j, a'_j)$ as the *head* and the pair $(i'_1, a'_1)$ as the *tail*.

We are now ready to describe the TTCC mechanism in its general form. Given the agents' stated preferences, the TTCC mechanism finds the outcome of the matching problem through the following procedure:

1. In each round of the TTCC algorithm:

   (i) based on her stated preferences, each agent $a_k$ who has not been assigned an item points at the best remaining unassigned item in $I_P \cup I_A$,
   (ii) each remaining agent who has been assigned an item continues to point at her assignment, and
   (iii) each remaining unassigned item in $I_P$ points to its owner.

2. Given the pointing in Step 1, there is either a cycle, or a chain; or both. At this step (Step 2), allocations associated with cycles are made:

(i) Proceed to Step 3 if there are no cycles. Otherwise, locate one cycle [7] and assign to each agent in the cycle the item that she points at. The assignment is final for these agents. *Remove all the agents and items in the cycle from the system.*

(ii) If every agent has been assigned an item, go to Step 4. Otherwise, go back to Step 1.

3. Each remaining item–agent pair initiates a chain. Since multiple chains may co-exist:

   (i) Select *only one* of the chains with a pre-defined chain selection rule. The assignment is final for the agents in the selected chain. In addition to selecting a chain, the chain selection rule also determines whether the selected chain is removed immediately or remains in the procedure.

   (ii) If every agent has been assigned an item, go to Step 4. Otherwise, go back to Step 1 to start the next round of allocation.

4. Remove all the chains (if there are any). The algorithm ends.

Intuitively, TTCC works in the following manner. It starts with agents' stated preferences. In each round of the algorithm, each agent who has not been assigned an item simply indicates (by "pointing") which one is her favorite among the remaining items. Given the "pointing" of the agents and the items, the algorithm identifies a cycle in the system and allocates items to agents in the cycle according to their "pointing". After the cycle is removed, the algorithm repeats from the first step because new cycles may form due to the "re-pointing" of agents and items. This process repeats until no cycle exists in the system. (Note that at this point all the agents and the items involved in the *cycles* have been removed from the system.)

Now each remaining agent initiates a chain, which, by our design, involves the lowest index at-large item among all the remaining at-large items. Since multiple chains may coexist, *a chain selection rule* is needed to pick a unique chain in a certain round. In [4], Roth et al. proposed several potential chain selection rules, among which chain selection rule ***e***— choose the chain starting with the highest priority agent and keep it in the system— is particularly interesting since it is the only rule proposed in [4] that makes TTCC both Pareto-efficient and strategy-proof. [8] Given *a fixed priority list* of the agents, in step 3(i) of TTCC, chain selection rule ***e*** requires that the unique chain whose tail is the highest priority agent (among the remaining agents) be chosen, and every agent in the selected chain obtain what they point at. Note that this selected chain is *not* removed from the system right away, so that the item at the tail (which belongs to the highest priority agent) becomes available for other agents in following rounds. After this step, the next round begins and the algorithm again repeats from the first step. TTCC ends when every agent gets an assignment from the set $I_P \cup I_A$.

[7] Recall that since cycles never intersect, when multiple cycles coexist, removing any one of them does not influence the others. Moreover, the order in removing cycles does not have any effect on the final outcome of the system.

[8] Other chain selection rules proposed by Roth et al. [4] include: Choose minimal chains and remove them; choose the longest chain and keep it; choose the chain starting with the highest priority agent and remove it, etc. In a more recent paper, Roth et al. [5] consider the more constrained case of pairwise kidney exchanges with 0–1 preferences and present constrained-efficient mechanisms that are strategy-proof.

We now proceed by presenting a simplified version of the A&S model, which is isomorphic to the above described kidney exchange model. Then we will see how A&S TTC can generate exactly the same outcome by changing "chains" into "cycles" so that they are identifiable by the A&S TTC mechanism.

## 4. A&S house allocation model

The model we present here is slightly different from the A&S house allocation model in [1]. We choose to use this simplified version of the A&S model so that the proof of our proposition is clear and straightforward. [9]

The model consists of:

1. a finite set of item–agent pairs [10] $\{(i_1, a_1), \ldots, (i_n, a_n)\}$ {We denote $I_P$ as the set of all items from the item–agent pairs, i.e., $I_P = \{i_1, \ldots, i_n\}$};
2. a finite set of $n$ at-large items $I_A = \{w_1, w_2, \ldots, w_n\}$;
3. for each agent $a_k$ ($k \in \{1, \ldots, n\}$), a strict preference relation $R_k$ over $I_P \cup I_A$, such that for all the agents:
   (i) $w_1 \succ w_2 \succ w_3 \succ \cdots \succ w_n$, and (ii) $\forall x \in I_P, x \succ w_n \Rightarrow x \succ w_j$ ($j \in \{1, 2, \ldots, n-1\}$) $\forall y \in I_P, y \prec w_1 \Rightarrow y \prec w_l$ ($l \in \{2, 3, \ldots, n\}$).

In words, again we have $n$ agents (applicants), each endowed with one indivisible item (the house). There are $n$ at-large items—$\{w_1, w_2, \ldots, w_n\}$, which represent the vacant houses that are available to the applicants in the allocation process. Each agent $a_k$ has a strict preference relation $R_k$ over all the items. This whole setting is a special (simpler) version of the original A&S house allocation model, where there is another set of new applicants who do not own any houses at the beginning of the algorithm. Besides, in the original A&S model the number of at-large items (vacant houses) can be different from the number of applicants—but here they are the same. Also, we impose restrictions (i) and (ii) on applicants' preference structure. Please note that these slight modifications not only do not influence the application of A&S TTC to the model, but also make our proof more straightforward.

Quite obviously, this house allocation model is isomorphic to the earlier kidney exchange model that we presented in Section 2.

## 5. A&S TTC mechanism

Given the above model, we now use A&S TTC mechanism to solve the matching problem. A&S TTC is also an algorithm consisting of multiple rounds. Similar to TTCC with chain selection rule $\boldsymbol{e}$, it starts with a *fixed priority list* of the agents and the agents' stated preferences.

---

[9] We thank two anonymous reviewers for making this suggestion.

[10] Here the term "agent" represents "applicant", and "item" represents "house".

### 5.1. Round 1

Define the set of *available items* for the first round to be $I_A$. Each agent $a_k$ points at her favorite item in $I_P \cup I_A$. Each item in $I_P$ points at the agent that it belongs to. Each available item points at the agent with highest priority. Since the numbers of agents and items are finite, there is at least one cycle. [11] The procedure will remove cycles one at a time as they form.

*Case* 1: When there exists a cycle that does not include an *available item* (i.e., the cycle is formed by agent–item pairs *only*), we choose this cycle and make allocations accordingly. (There could be multiple cycles that do not include an available item, in which case we simply pick one of them. [12] ) Every agent who participates in this cycle is assigned the item that she points at and removed from the system with her assignment.

*Note*: If there is such a cycle under A&S TTC, the same cycle would have formed under TTCC, since in the first round under both algorithms each agent points at her favorite item and each item in $I_P$ points at its owner. Moreover, we would have made the same allocation and removed the cycle under TTCC.

*Case* 2: If such a cycle (i.e., cycle formed by agent–item pairs *only*) does not exist then there is a unique cycle. [13] In particular this cycle includes $w_1$ and the highest priority agent. A&S TTC identifies this unique cycle, assigns every agent in the cycle the item that she points at, and removes the agent with her assignment. However, the item that was owned by the highest priority agent *remains* since it is not in the cycle and it has not been assigned to any agent. This item is added to the set of *available items* for *Round* 2. All *available items* that are not removed (i.e., $w_2, \ldots, w_n$) remain in the set of *available items*. If there is at least one remaining agent then we go to the next round.

*Note*: The unique cycle identified in Case 2 would have formed as a chain under the TTCC algorithm, and in particular, it is the chain selected by rule ***e*** in step 3(i). $w_1$ would be the first waitlist option assigned to one of the agents. The only difference is that the selected chain would have been kept in the system under TTCC (with rule ***e***) while this cycle is removed under A&S TTC. This, however, does not generate any difference in final outcomes because the only reason for keeping the selected chain under TTCC is to make use of the unassigned item at the "tail" of the chain in later rounds. (Recall that all the assignment associated with the chain is final although the agents and items remain in the system.) The unassigned item at the tail of the chain is exactly the item that is added to the set of available items under A&S TTC. Therefore, keeping the whole chain and fixing the assignments of involved agents (in TTCC with rule ***e***) is equivalent to removing the cycle but adding the unassigned item to the set of available items (in A&S TTC).

[11] Here a *cycle* does *not* have to be an ordered list of *item–agent pairs*. Instead, consistent with [1], it is defined as an ordered list of agents and items $(j_1, j_2, \ldots, j_k)$ where $j_1$ points at $j_2$, $j_2$ points at $j_3, \ldots,$ and $j_k$ points at $j_1$. In specific, one of the components in this list of agents and items can be one of the *available items*.

[12] Similar to TTCC, under A&S TTC the coexisting cycles never intersect. Thus the order of removing cycles does not affect final outcomes.

[13] Recall that we assume for all agents, their preference over at-large items is such that $w_1 \succ w_2 \succ w_3 \succ \cdots \succ w_n$. Therefore, if any at-large item is in a cycle in round 1, it has to be $w_1$. In this case, this is the only cycle that involves an at-large item, and the highest priority agent is also part of the cycle.

### 5.2. Round $t$

The set of *available items* for *Round* $t$ is defined at the end of *Round* $t-1$. Each remaining agent points at her favorite item among the unassigned items in $I_P \cup I_A$. Each item in remaining item–agent pairs points at the agent it belongs to. Each *available item* points at the agent with highest priority among the remaining agents. There is at least one cycle. At this point:

*Case* 1: If multiple cycles co-exist, choose a cycle without any *available item* in it. Every agent in this cycle is assigned the item that she points at and removed from the system with her assignment.

*Case* 2: If such a cycle does not exist then there is a unique cycle. In particular this cycle includes the highest priority agent (among the remaining agents) and one of the *available items*. This available item is either the lowest index at-large item among the remaining at-large items, or an item that was originally owned by an agent but was added to the set of *available items* in an earlier round. A&S TTC identifies this unique cycle, assigns every agent in the cycle the item that she points at, and removes the agent with her assignment. The item that was owned by the highest priority agent is added to the set of *available items* for *Round* $t+1$. All *available items* that are not removed remain available. If there is at least one remaining agent then we go to the next round.

*Note*: In Round 1, A&S TTC and TTCC (with rule $\boldsymbol{e}$) make exactly the same assignment. Therefore, for both algorithms Round 2 starts with the same set of remaining agents and the same set of unassigned items. Given the newly-defined sets, we repeat the procedure which again gives the same assignment under the two mechanisms. By repetition we know A&S TTC and TTCC (with rule $\boldsymbol{e}$) generate the same outcome.

Intuitively, by viewing waitlist options as available items and letting them point at the highest priority agent, A&S TTC algorithm can identify not only all the cycles formed by agent–item pairs (as TTCC does), but also the unique "chain" that is selected by chain selection rule $\boldsymbol{e}$ in [4]. This guarantees that A&S TTC mechanism and TTCC mechanism generate the same outcome given a fixed list of priority of the agents and the agents' stated preferences.

## 6. Conclusions

As concluding remarks, we emphasize that TTCC and A&S TTC are equivalent only when chain selection rule $\boldsymbol{e}$ in [4] is used for picking chains under TTCC. TTCC can be implemented with a variety of other chain selection rules and our results do not carry over for these other rules.

We prove the equivalence between A&S TTC and TTCC simply by viewing the waitlist option $w$ as at-large items that can be assigned to agents and generalizing the concept of cycles so that A&S TTC can identify the selected chains as well. Thus, we show that the A&S TTC algorithm can be used to handle a problem with waitlist options, which need to be solved by TTCC otherwise. In addition, since it has been well established that A&S TTC as a matching mechanism is individually rational, Pareto-efficient and strategy-proof, these theoretical properties automatically transfer to TTCC with chain selection rule $\boldsymbol{e}$. This is consistent with the theoretical findings of [4] where Roth et al. show these properties to hold for TTCC when rule ***e*** is used.

Theoretically, our work establishes the link between two important matching mechanisms: the A&S Top Trading Cycles mechanism and the Top Trading Cycles and Chains mechanism. While the equivalence observed is relevant for one specific case of the TTCC mechanism, it is a particularly interesting case since it is the only version identified by Roth et al. [4] to be both Pareto-efficient and strategy-proof. Research on both problems is clearly of significant practical importance. We believe our result has important theoretical and applied mechanism design implications.

## Acknowledgments

The authors would like to thank Yan Chen, Tayfun Sonmez, and M. Utku Unver for their helpful comments on earlier versions of this paper.

## References

[1] A. Abdulkadiroglu, T. Sonmez, House allocation with existing tenants, J. Econ. Theory 88 (1999) 233–260.
[2] Y. Chen, T. Sonmez, Improving efficiency of on-campus housing: an experimental study, Amer. Econ. Rev. 92 (5) (December 2002) 1669–1689.
[3] A.E. Roth, A. Postlewaite, Weak versus strong domination in a market with indivisible goods, J. Math. Econ. 4 (2) (August 1977) 131–137.
[4] A.E. Roth, T. Sonmez, M.U. Unver, Kidney exchange, Quart. J. Econ. 119 (2) (May 2004) 457–488.
[5] A.E. Roth, T. Sonmez, M.U. Unver, Pairwise kidney exchange, J. Econ. Theory, forthcoming.
[6] L. Shapley, H. Scarf, On cores and indivisibility, J. Math. Econ. 1 (1974) 23–37.
[7] T. Sonmez, M.U. Unver, House allocation with existing tenants: an equivalence, Games Econ. Behav. 52 (1) (July 2005) 153–185.
[8] Y. Wang, A. Krishna, Timeshare allocation mechanisms, working paper, University of Michigan, 2005.