Housing Markets with Indifferences: a Tale of Two Mechanisms

Haris Aziz and Bart de Keijzer

Abstract

The (Shapley-Scarf) housing market is a well-studied and fundamental model of an exchange economy. Each agent owns a single house and the goal is to reallocate the houses to the agents in a mutually beneficial and stable manner. Recently, Alcalde-Unzu and Molis [2011] and Jaramillo and Manjunath [2011] independently examined housing markets in which agents can express indifferences among houses.They proposed two important families of mechanisms, known as TTAS and TCR respectively. We formulate a family of mechanisms which not only includes TTAS and TCR but also satisfies many desirable properties of both families. As a corollary, we show that TCR is strict core selecting (if the strict core is non-empty). Finally, we settle an open question regarding the computational complexity of the TTAS mechanism. Our study also raises a number of interesting research questions.

1 Introduction

Housing markets are fundamental models of exchange economies of goods where the goods could range from dormitories to kidneys [Sönmez and Ünver, 2011]. The classic housing market (also called the Shapley-Scarf Market) consists of a set of agents each of which owns a house and has strict preferences over the set of all houses. The goal is to redistribute the houses to the agents in the most desirable fashion. Shapley and Scarf [1974] showed that a simple yet elegant mechanism called *Gale's Top Trading Cycle (TTC)* is strategy-proof and finds an allocation which is in the core. TTC is based on multi-way exchanges of houses between agents. Since the basic assumption in the model is that agents have strict preferences over houses, TTC is also strict core selecting and therefore Pareto optimal.

Indifferences in preferences are not only a natural relaxation but are also a practical reality in many cases. Many new challenges arise in the presence of indifferences: core stability does not imply Pareto optimality; the strict core can be empty [Quint and Wako, 2004]; and strategic issues need to be re-examined. In spite of these challenges, Alcalde-Unzu and Molis [2011] and Jaramillo and Manjunath [2011] proposed desirable mechanisms for housing markets with indifferences. Alcalde-Unzu and Molis [2011] presented the *Top Trading Absorbing Sets (TTAS)* family of mechanisms which are strategy-proof, core selecting (and therefore individually rational), Pareto optimal, and strict core selecting (if the strict core is non-empty). Independently, Jaramillo and Manjunath [2011] came up with a different family of mechanisms called *Top Cycle Rules (TCR)* which are strategy-proof, core selecting, and Pareto optimal. Whereas it was shown in [Jaramillo and Manjunath, 2011] that each TCR mechanism runs in polynomial time, the time complexity of TTAS was raised as an open problem in [Alcalde-Unzu and Molis, 2011].

We first highlight the commonality of TCR and TTAS by describing a simple class of mechanisms called *Generalized Absorbing Top Trading Cycle (GATTC)* which encapsulates the TTAS and TCR families. It is proved that each GATTC mechanism is core selecting, strict core selecting, and Pareto optimal. As a corollary, TCR is strict core selecting. We note that whereas a GATTC mechanism satisfies a number of desirable properties, the strategy-proofness of a particular GATTC mechanism hinges critically on the order and way of choosing trading cycles. Finally, we settle the computational complexity of TTAS.

By simulating a binary counter, it is shown that a TTAS mechanism can take exponential time to terminate.

2 Preliminaries

Let N be a set of n agents and H a set of n houses. The endowment function $\omega : N \rightarrow H$ assigns to each agent the house he originally owns. Each agent has complete and transitive preferences $\succsim_i$ over the houses and $\succsim = (\succsim_1, \ldots, \succsim_n)$ is the preference profile of the agents. The *housing market* is a quadruple $M = (N, H, \omega, \succsim)$. For $S \subseteq N$, we denote $\omega(S) = \{\omega(i) : i \in S\}$ by $\omega(S)$. A function $x : S \rightarrow H$ is an *allocation* on $S \subseteq N$ if there exists a bijection π on S such that $x(i) = \omega(\pi(i))$ for each $i \in S$. The goal in housing markets is to re-allocate the houses in a mutually beneficial and efficient way. An allocation is *individually rational (IR)* if $x(i) \succsim_i \omega(i)$. A coalition $S \subseteq N$ *blocks* an allocation x on N if there exists an allocation y on S such that for all $i \in S$, $y(i) \in \omega(S)$ and $y(i) \succ_i x(i)$. An allocation x on N is in the *core (C)* of market M if it admits no blocking coalition. An allocation that is in the core is also said to be *core stable*. An allocation is *weakly Pareto optimal (w-PO)* if N is not a blocking coalition. A coalition $S \subseteq N$ *weakly blocks* an allocation x on N if there exists an allocation y on S such that for all $i \in S$, $y(i) \in \omega(S)$, $y(i) \succsim_i x(i)$, and there exists an $i \in S$ such that $y(i) \succ_i x(i)$. An allocation x on N is in the *strict core (SC)* of market M if it admits no weakly blocking coalition. An allocation that is in the strict core is also said to be *strict core stable*. An allocation is *Pareto optimal (PO)* if N is not a weakly blocking coalition. It is clear that strict core implies core and also Pareto optimality. Core implies weak Pareto optimality and also individual rationality.

A mechanism that always returns a Pareto optimal and (strict) core stable allocation is said to be *Pareto optimal* and *(strict) core-selecting* respectively. A mechanism is *strategy-proof* if for each agent, reporting false preferences to the mechanism will not be beneficial to the agent (i.e., when the agent reports false preferences, he will not end up with a house that he prefers more than the house he would get when he reports his true preferences to the mechanism).

Desirable allocations of housing markets can be computed via a graph-theoretic approach. Each housing market $M = (N, H, \omega, \succsim)$ has a corresponding simple digraph $G(\succsim) = (N \cup H, E)$ such that for each $i \in N$ and $h \in H$, $(i, h) \in E$ if $h \succsim h'$ for all $h' \in H$, and (h, i) if $h = \omega(i)$. In other words, each agent points to his maximally preferred houses and each house points to his owner. An *absorbing set* of a digraph is a strongly connected component from which there are no outgoing edges. Two nodes constitute a *symmetric pair* if there are edges from each node to the other. Both nodes are then called *paired-symmetric*. An absorbing set is *paired-symmetric* if each node belongs to a symmetric pair.

3 GATTC

In this section, we formulate a simple family of mechanisms called *Generalized Absorbing Top Trading Cycle (GATTC)* which is designed for housing markets with indifferences and extends not only TTC but also includes the two families TTAS and TCR. It is based on multi-way exchanges of houses between agents. We will show that GATTC satisfies many desirable properties of housing mechanisms such as being core-selecting and Pareto optimal.

Before we describe GATTC, we will introduce the original TTC mechanism which is for the domain of housing markets with strict preferences. TTC works as follows. For a housing market M with strict preferences, we first construct the corresponding graph $G(\succsim)$ as defined above. Then, we start from an agent and walk arbitrarily along the edges until a cycle is completed. A cycle starting from any agent is of course guaranteed to exist as each node in $G(\succsim)$ has positive outdegree. This cycle is removed from $G(\succsim)$. Within the removed cycle, each agent gets the house he was pointing to. The graph $G(\succsim)$ is *adjusted* so that the remaining agents point to the most preferred houses among the remaining houses. The process is repeated until all the houses and agents are deleted from the graph.[1]

For a housing market with indifferences, TTC can still be used to return a core selecting allocation: break ties arbitrarily and then run TTC [see *e.g.*, Ehlers, 2012]. However such an allocation may not be Pareto optimal [see *e.g.*, Alcalde-Unzu and Molis, 2011, Jaramillo and Manjunath, 2011]. GATTC achieves Pareto optimality and is based on absorbing sets and the concept of a 'good cycle'. A *good cycle* is any cycle in $G(\succsim)$ which contains at least one node that is not paired-symmetric. By *implementing a cycle* we mean reallocating the houses along the cycle. For example consider the cycle $a_0, h_1, a_1, \ldots, h_m, a_m, h_0, a_0$. Then for all $i \in \{0, \ldots, m\}$, house $h_{i+1 \mod m}$ is made to point to a_i. The following is the description of a GATTC mechanism.

GATTC

`Let` $G = G(\succsim)$ `and repeat the following until` G `is empty.`

1. `Repeat the following a finite number of times on` G`:`
   - 1.1. `Either implement a non-good cycle (if` G `is not empty), or do nothing.`
   - 1.2. `Either remove a paired-symmetric absorbing set and adjust`[2] G `(if a paired-symmetric absorbing set exists), or do nothing.`
2. `Repeatedly remove paired-symmetric absorbing sets and adjust` G`, until there are no paired-symmetric absorbing sets in` G.
3. `If` G `is not empty, implement a good cycle.`

We stress that the choices that a GATTC mechanism makes in steps 1.1. and 1.2. are allowed to be different each time the mechanism executes these steps during the same run. The same holds for the number of times that steps 1.1. and 1.2. are repeated, each time that step 1 is executed. It is not even required that a GATTC mechanism is deterministic: as long as it has the property that the output can always be obtained by a procedure that respects the form above, it is part of the GATTC family.

Example 1 *Consider a housing market* $M = (N, H, \omega, \succsim)$ *where* $N = \{a_1, \ldots, a_5\}$, $H = \{h_1, \ldots, h_5\}$, ω *is such that* $\omega(a_i) = h_i$ *for all* $i \in \{1, \ldots, 5\}$, *and preferences* $\succsim$ *are defined as follows:*

agent	a_1	a_2	a_3	a_4	a_5
preferences	h_2	h_3	h_4, h_5	h_1	h_2
	h_1	h_2	h_3	h_5	h_4
				h_4	h_5

Then, if ties are broken in any way, TTC does not return a Pareto optimal allocation. However, GATTC (or TTAS/TCR) returns the following Pareto optimal allocations: $\{\{a_1, h_2\}, \{a_2, h_3\}, \{a_3, h_5\}, \{a_4, h_1\}, \{a_5, h_4\}\}$ *or* $\{\{a_1, h_1\}, \{a_2, h_3\}, \{a_3, h_4\}, \{a_4, h_5\}, \{a_5, h_2\}\}$. *Figure 2 (placed at the end of this paper, due to space constraints) illustrates the first steps in the execution of a GATTC mechanism on this housing market.*

Illustration of the first steps of a GATTC mechanism applied to the housing market in Example 1.

[1] Please see Section 2.2 of [Sönmez and Ünver, 2011] for an elegant illustration of how TTC works.

[2] *Adjusting* is defined here in the same way as for the TTC mechanism.

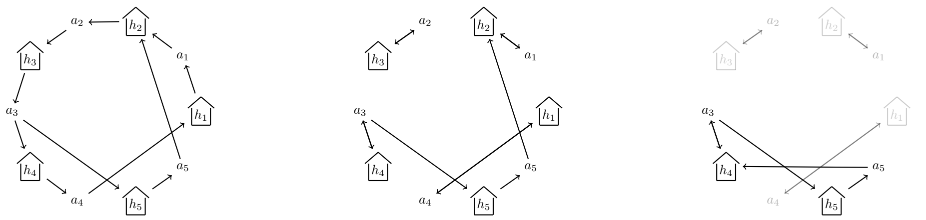

Figure 1: Illustration of the first steps of a GATTC mechanism applied to the housing market in Example 1. The top figure shows the graph as initialized. The algorithm proceeds by executing step 1 zero times, removing no paired-symmetric absorbing sets in step 2 (as there are none), and implementing the cycle $(a_1, h_2, a_2, h_3, a_3, h_4, a_4, h_1, a_1)$ in step 3. The graph after implementing this cycle is shown in the middle figure. Subsequently, the mechanism removes the paired-symmetric absorbing sets, forcing a_5 to point to his second-most preferred houses, i.e., house h_4.

We say that a housing market mechanism is *valid* if it terminates and returns a proper allocation.

Theorem 1 *GATTC is valid, core-selecting, and Pareto optimal.*

Proof: We prove each property separately:

- *Valid:* At the beginning of every step, G has the property that each node has positive out-degree. For non-empty graphs with this property, an absorbing set of cardinality greater than 1 is guaranteed to exist [Kalai and Schmeidler, 1977]. Therefore, if G is not empty, then at step 1.1. there is guaranteed to be a cycle, and at step 3. there is guaranteed to be a good cycle (because there must be an absorbing set that is not paired-symmetric). In each iteration (of steps 1, 2, and 3), if paired-symmetric absorbing sets exist they are removed in Step 2.[3] Also, at least one good cycle is implemented in step 3 which reduces the number of non-paired-symmetric nodes. Therefore, there can be a maximum of $O(n)$ iterations until GATTC terminates. Since each removed house is allocated to the agent it was last pointing to, GATTC returns a proper allocation.
- *Core selecting:* When any agent i is removed from the graph along with his allocated house h, then h is a maximal house for i from among the remaining houses. Therefore i cannot be in a blocking coalition with the agents remaining in the graph.
- *Pareto optimal*: Let S_k be the kth paired-symmetric absorbing set that arises at some point in the GATTC mechanism (and is thus removed from the graph by the GATTC mechanism, and is included accordingly in the allocation produced by the GATTC mechanism). In any allocation x in which none of the players in S_1 are worse off than in the allocation produced by GATTC, the players in S_1 must be allocated to houses in S_1. Taking this as the base case, it follows by easy induction that in x, the players of S_k must be allocated to houses in the kth paired-symmetric absorbing set. Next, suppose that i is a player in S_k for some k. Then no house in S_k is more preferred by i than the house that the GATTC mechanism assigns him to. It follows that no player is strictly better off in x than in the allocation produced by GATTC.

This completes the proof. □

[3] An absorbing set of a graph can be computed in linear time via the algorithm of Tarjan [1972].

Theorem 2 *GATTC is strict core selecting in case the strict core is non-empty.*

Proof: We prove the statement by proving two claims.

Claim 1 *GATTC ensures that if each agent in an absorbing set A can get his maximal house within A, then it will.*

Proof: Define an *inward set* as a set of vertices without edges pointing outward from A. An absorbing set is by definition an inward set. We prove this claim for the more general notion of inward sets. Let A be an inward set that arises at some point in time t during execution of the GATTC mechanism, and assume that each agent can simultaneously get a maximal house in A. If A eventually becomes paired-symmetric, then every agent in A surely gets a maximal house within A. Let us thus assume that A does not eventually become paired-symmetric. Consider the first point in time t' where vertices are removed from A by the mechanism. This point t' exists because the mechanism terminates. All cycles that are implemented in between t and t' either lie completely inside A or completely outside A, because there are no edges pointing from outside A to a vertex in A. It follows that at point t', the removed paired-symmetric absorbing set A' is a strict subset of A. Note that agents in $A \setminus A'$ cannot get a house from within A' without some agent in A' getting a worse house. Hence, by the assumption that each agent in A can get his maximal house within A, it follows that agents in $A \setminus A'$ can still all get a maximal house from within $A \setminus A'$. The proof follows by induction; repeating the same argument on the inward set $A \setminus A'$ that arises when removing A' from the graph. □

Claim 2 *The returned allocation x is in the strict core if and only if for each absorbing set A encountered in the algorithm, each agent in A will get his maximal house in A.*

Proof: ($\Rightarrow$) Assume there is an agent $i \in A$ such that there exists a house h in A for which $h \succsim_i x(i)$. But then i can be involved in a weakly blocking coalition by forming a cycle within A.

($\Leftarrow$) Assume that each agent i in A gets a maximal house from within A. Thus i cannot be part of a blocking coalition. It could still be part of a weakly blocking coalition if an agent i in A had a maximal house h outside A within the remaining graph and there exists a cycle of the form $i, h, \ldots, i$. But this is not possible since A is absorbing. □

From the two claims, the theorem follows. □

We also observe that on the domain of strict preferences, GATTC is equivalent to TTC. The reason is that implementation of any cycle results in a paired-symmetric absorbing set which is then removed from the graph. Ma [1994] proved that for housing markets with strict preferences, a mechanism is strict core selecting if and only if it is individually rational, Pareto optimal, and strategy-proof. On the other hand, we note that in the presence of ties, even if a mechanism is (strict) core selecting, and Pareto optimal, it is not necessarily strategy-proof.

Theorem 3 *Not every GATTC mechanism is strategy-proof.*

Proof Sketch: Consider the following GATTC mechanism in which no non-good cycle is implemented and every good cycle is found in the following way. Consider $a_i \in N$, $h_j \in H$ such that $(a_i, h_j) \in E$, $(h_j, a_i) \notin E$, and a_i and h_j are in a strongly connected component. Then, there exists a shortest path P from h_j to a_i. Find this path P by Dijkstra's shortest path algorithm. Path P gives us a good cycle a_i, h_j, P, a_i.

For this subclass of GATTC, it can be shown that an agent may have incentive to lie about his preferences to obtain a better allocation. Informally, there exist instances of a housing market in which if an agent a does not lie, it may only get a third most preferred house. However, if a points to his second most preferred house h in the graph, it can manage to influence which good cycle is selected and be included in that good cycle. Agent a then gets allocated h. □

4 TTAS and TCR

We now describe the two families of mechanisms in the literature — TTAS [Alcalde-Unzu and Molis, 2011] and TCR [Jaramillo and Manjunath, 2011] — designed for housing markets with indifferences. Both families of mechanisms are extensions of TTC. We will later show that both families are subclasses of GATTC.

TTAS

Fix a priority ranking of the houses; i.e., a complete, transitive and antisymmetric binary relation over H. Construct the graph $G(\succsim)$, and run the following procedure on it (starting with $i = 1$, incrementing i every iteration) until no more agents are remaining in the graph.

Step i

- (i.1) Let each remaining agent point to her maximal houses among the remaining ones. Select the absorbing sets of this digraph.
- (i.2) Consider the paired-symmetric absorbing sets. Their agents are allocated the house that the agents currently point to in the graph. These absorbing sets are removed from the graph.
- (i.3) Consider the remaining absorbing sets. Select for each agent a unique house to point to by using the following criterion: each agent i currently owning house h provisionally points only to the house that i likes most (according to $\succsim_i$) among the houses remaining. Ties are broken by selecting among the candidate houses the one that comes after h in the priority order (if there is no such house, then select among the candidate houses the first house in the priority order).
- (i.4) Then, in this subgraph, there is necessarily at least one cycle and no two cycles intersect. Assign (provisionally) to each agent in these cycles the house that he is pointing to, but do not remove them from the graph.

The algorithm terminates when no agents and houses remain, and the outcome is the assignment formed during its execution.

TCR

Consider a priority ranking of the agents; i.e., a complete, transitive and antisymmetric binary relation over A. Do the following until no more agents are left.

1. Departure: A group of agents is chosen to ‘‘depart’’ if two conditions are met: i) What each agent in the group holds is among his most preferred houses (among the remaining ones), and ii) All of the most preferred houses (among the remaining ones) of the group are held by them. Once a group departs, each agent in it is assigned what he holds and is removed from the set of remaining agents. In addition, their houses are removed from the remaining houses. There may be another group that can be chosen to depart.

```
   The process continues until there are no more groups that can depart.  If the
   two conditions are not met by any group, then nobody departs.

2. Pointing:  Each agent points to an agent holding one of his top houses (among
   the remaining ones).  Since there may be more than one such agent, the problem
   of figuring whom each agent points to is a complicated one.

   We solve it in stages as follows:

Stage 1 For each remaining j such that j holds the same house that he held in the
        previous step, each i that pointed at j in the previous step points to j
        in the current step.  Of course, this does not apply for the very first
        step.

Stage 2 Each i with a unique top house (among the remaining ones) points to the
        agent holding it.

Stage 3 Each agent who has at least one of his top houses (among the remaining
        ones) held by an unsatisfied agent points to whomever has the highest
        priority among such unsatisfied agents.

Stage 4 Each agent who has at least one of his top houses (among the remaining
        ones) held by a satisfied agent who points to an unsatisfied agent points
        to whomever points to the unsatisfied agent with highest priority.  If
        two or more of his satisfied ``candidates'' point to the unsatisfied
        agent with highest priority, he points to the satisfied candidate with the
        highest priority.

Stage ... And so on.

3. Trading:  Since each remaining agent points to someone, there is at least
   one cycle of remaining agents.  For each such cycle, agents trade according
   to the way that they point and what they hold for the next step is updated
   accordingly.
```

Note that TTAS and TCR mechanisms depend on the priority ordering over H and A respectively. The variation in priority rankings leads to classes of mechanisms rather than a single mechanism. Next, we show that TTAS and TCR are subclasses of GATTC in which cycles are selected via the strict order over houses and agents respectively.

Theorem 4 *GATTC generalizes both the TTAS and TCR families of mechanisms.*

Proof: (*GATTC generalizes TTAS*). (*GATTC generalizes TTAS*). Step i.2 of TTAS corresponds to repeatedly executing step 1.2. (and skipping step 1.1). After that, TTAS may implement a number of non-good cycles. This corresponds in GATTC to executing step 1.1 (skipping step 1.2). However, the proof of Proposition 1 in [Alcalde-Unzu and Molis, 2011] shows that TTAS can never perpetually implement non-good cycles: Either the graph becomes empty, or eventually a good cycle is found and implemented. So executing in TTAS step i.2 to i.4 on iterations where a good cycle is implemented, corresponds to executing steps 3 and 4 of GATTC.

(*GATTC generalizes TCR*). A TCR rule reduces to the GATTC mechanism if zero non-good cycles are implemented in Step 1. and if in Step 3 of GATTC, a good cycle is implemented in the particular way as outlined in the definition as TCR. It is clear from the Step 2 (pointing) of TCR that the way agents are made to point, the cycle induced involves at least one node which is not paired-symmetric. Therefore the cycle in question is a good cycle. □

In contrast to TTAS (which is strict core-selecting), it was not known whether TCR is also strict core-selecting. As a corollary of Theorems 2 and 4, we obtain the following.

Corollary 1 *Each TCR mechanism is strict core selecting (if the strict core is non-empty).*

In the next section, we answer an open question concerning the running time of the TTAS mechanism.

5 Complexity of TTAS

An important property of TTAS is that if an agent i is reallocated a house h during the running of TTAS but i and h are not yet deleted from the graph, then agent i is guaranteed to be finally allocated a house $h' \in H$ such that $h \sim_i h'$ [Lemma 1, Alcalde-Unzu and Molis, 2011]. Therefore the number of symmetric pairs can only increase during the running of the algorithm although it may stay constant in a number of iterations. Alcalde-Unzu and Molis [2011] showed that despite a number of stages in which no obvious progress is being made, TTAS eventually terminates [Proposition 1, Alcalde-Unzu and Molis, 2011]. Although, we know that TTAS terminates and results in a proper allocation, the proof of [Proposition 1, Alcalde-Unzu and Molis, 2011] does not help shed light on how many steps are taken in the running of TTAS.We will show the following.

Theorem 5 *There exists a family of housing markets $\{M_k = (N_k, H_k, \omega_k, \preceq^k) : k \in \mathbb{N}_{>0}\}$ with $|N_k| = |H_k| = 2k+1$, and corresponding priority rankings $\{R_k : k \in \mathbb{N}_{>0}\}$ such that if the TTAS mechanism receives input M_k and chooses R_k as its priority ranking in step 0, then the TTAS mechanism runs for at least $2^k = 2^{(|N_k|-1)/2}$ steps until it terminates.*

This theorem shows thus that the TTAS mechanism, according to its current description, does not run in polynomial time. It still might be that for each instance, there is some priority ranking such that the TTAS mechanism runs in polynomial time, but then at least some additional details are needed in the description on how to choose the priority ranking. The algorithm described in Alcalde-Unzu and Molis [2011] is not sufficient.

Proof: The houses and agents of housing market M_k are named as $H_k = \{h_1, h'_1, h_2, h'_2, \ldots, h_k, h'_k, h_{k+1}\}$ and $\{a_1, a'_1, a_2, a'_2, \ldots, a_k, a'_k, a_{k+1}\}$ respectively. In the initial endowment, house h_j is assigned to agent a_j for all $j \in [k+1]$,[4] and house h'_j is assigned to agent a'_j for all $j \in [k]$. The preference profile of agent a_j, $j \in [k]$ is described by two equivalence classes: his class of most preferred houses is $\{h'_j, h_j, h_{j+1}\}$, and the remainder of the houses is in his other equivalence class, i.e., his class of least preferred houses. The preference profile of agent a'_j, $j \in [k]$, is also described by two equivalence classes: His class of most preferred houses is $\{h_j, h'_j, h_1\}$ (so for $j = 1$, this set has cardinality 2), and the remainder of the houses are in the other equivalence class, i.e., his class of least preferred houses. The preference profile of agent a_{k+1} is also described by two equivalence classes: His class of most preferred houses is $\{h_1\}$, and the remainder of the houses is in his other equivalence class, i.e., his class of least preferred houses. The priority ranking R is $(h_1, h'_1, h_2, h'_2, \ldots, h_k, h'_k, h_{k+1})$.

The high level idea of this example is to simulate a binary counter. The graph that the TTAS mechanism maintains will contain a single absorbing set at every step: the entire graph. At every step except the last one, the only agent that prevents the graph from being paired-symmetric will be agent a_{k+1}. We associate bit-strings of length k to the graphs that may arise in some of the steps of the TTAS algorithm: Let $b \in \{0,1\}^k$ be any bit-string of length k, then we define the graph G_b as the graph where for all j,

- a_j and a'_j all point to their set of most preferred houses,

[4] Suppose $x \in \mathbb{N}$, then $[x]$ stands for the set $\{1, \ldots, x\}$.

- if $b_j = 0$, then h_j points to a_j and h'_j points to a'_j.
- if $b_j = 1$, then h_j points to a'_j and h'_j points to a_j.

We prove that for all bit-strings b of length k there is a step i_b such that the graph at the beginning of step i_b is equal to G_b. Because there are 2^k possible bit-strings, it then follows that there are at least 2^k steps before the algorithm terminates.

In order to understand what happens during the execution of the TTAS algorithm on an instance M_j, it will be helpful to look at the example of Figure 2, where the graph at the beginning of every step is shown when we run the TTAS mechanism on M_3.

Let us assume that at the beginning of step i of the execution of the TTAS mechanism, the graph is equal to G_b for some b. We can prove that G_b is strongly connected:

Claim 3 *For each length k bit-string b, G_b is strongly connected.*

Proof: We first show that there is a path from h_1 to every other vertex v.

If $b_1 = 0$, then h_1 points to a_1 and h'_1 points to a'_1. If $b_2 = 0$, then there exists a path $(h_1, a_1, h_2, a_2, h'_2, a'_2)$. If $b_2 = 1$, then there exists a path $(h_1, a_1, h_2, a_2, h'_2, a_2)$.

If $b_1 = 1$, then h_1 points to a'_1 and h'_1 points to a_1. If $b_2 = 0$, then there exists a path $(h_1, a'_1, h'_1, a_1, h_2, a_2, h'_2, a'_2)$. If $b_2 = 1$, then there is a path $(h_1, a'_1, h'_1, a_1, h_2, a'_2, h'_2, a_2)$.

Therefore h_1 has a path to each of the following vertices: $a_1, a_2, h_1, h_2, a'_1, a'_2, h'_1, h'_2$.

Using the same argument, we can see that for each a_j, there is a path to a_{j+1}; for each a'_j, there is a path to a'_{j+1}; for each h_j there is a path to h_{j+1}; for each h'_j, there is a path to h'_{j+1}. Therefore, it holds that: From h_1, there is a path to each a_j for $j \in [k+1]$; From h_1, there is a path to each a'_j for $j \in [k]$; From h_1, there is a path to each h_j for $j \in [k+1]$; and from h_1, there is a path to each h'_j for $j \in [k]$.

Similarly, it can be shown that from every vertex, there is a path to h_1. This completes the argument of the claim.

□

Therefore, G_b has only one absorbing set: the whole of G_b.

Also observe that for all b, G_b is not paired symmetric, because of player $k+1$. From this we conclude that if the graph at the beginning of a step i is equal to G_b, for some $b \in \{0,1\}^k$, then the TTAS mechanism does not terminate at step i, and the mechanism will certainly reach step $i+1$.

For some step i of the TTAS mechanism, and for every agent $a \in N$, let S^i_a denote the set of most preferred houses of a that are ranked lower than the house assigned to a in step i. However, if this set is empty, then define S^i_a to be the set of most preferred houses of a. Let us assume that for step i, the following property holds, which we will call *Property A_i*: for every agent $a \in N$, it holds that the set of most preferred houses of a that have been provisionally assigned to a the least number of times (including 0 times), is S^i_a.

We define a straightforward bijection $c : \{0,1\}^k \to [2^k - 1] \cup \{0\}$ as follows: bit-string b corresponds to the integer $\sum_{j=1}^{k} 2^{j-1} b_j$. We then see that the following happens:

Claim 4 *Let b be a bit-string of length k, suppose that i is a step in the TTAS mechanism such that the graph at step i is equal to G_b, and suppose that Property A_i holds.*

- *If $c(b)$ is even, then the graph at step $i+1$ of the TTAS algorithm is equal to G_{b+1}, and Property A_{i+1} holds.*
- *If $c(b)$ is odd and not equal to $2^k - 1$, then the graph at step $i+2$ of the TTAS algorithm is equal to G_{b+1}, and Property A_{i+2} holds.*

Proof: If $c(b)$ is even, it is easy to see that at the beginning of step $i+1$, the graph will be $G_{c^{-1}(c(b)+1)}$: the only cycle found in part 3 of step i is $(h_1, a_1, h'_1, a'_1, h_1)$. Any other cycles would have to make use of one of the arcs pointing toward h'_1, but that is not possible by the vertex-disjointness property of the cycles in the subgraph used at part 3 of step i. After augmenting G_b according to cycle $(h_1, a_1, h'_1, a'_1, h_1)$, it is easy to check that the graph is equal to G_{b+1}. Also, observe that Property A_{i+1} holds.

If $c(b)$ is odd and not equal to $2^k - 1$, then define j to be the largest index such that $b_{j'} = 1$ for all $j' \leq j$. Then, in part 3 of step i, the cycle $(h_1, a'_1, h'_1, a_1, h_2, a'_2, h'_2, a_2, \ldots, h_j, a'_j, h'_j, a_j, h_{j+1}, a_{j+1}, h'_{j+1}, a'_{j+1}, h_1)$ is found, and no other cycle is found, because otherwise h_1 would be in such a cycle: a contradiction. It is not hard to verify that property A_{i+1} holds, and the graph that now arises at the beginning of step $i+1$ is again a single absorbing set that is not paired symmetric, because of a_{k+1}. Step $i+2$ will therefore certainly be reached, and it can be verified by similar reasoning as before that again a single cycle is found in part 3 of step $i+1$. This cycle is $(h_1, a'_{j+1}, h_{j+1}, a_j, h_j, a_{j-1}, h_{j-1}, a_{j-2}, h_{j-2}, \ldots, a_1, h_1)$. Augmenting the graph on this cycle makes the graph exactly equal to $G_{c^{-1}(c(b)+1)}$. Moreover, Property A_{i+2} holds. □

Property A_1 is certainly satisfied, and the graph at step 1 is $G_{000\ldots}$. By straightforward induction, using the claim above, it follows that for all bit-strings b of length k there is indeed a step i_b such that the graph at the beginning of step i_b is equal to G_b. □

6 Discussion

Properties	TTAS	TCR	GATTC
Core, Pareto optimal	✓	✓	**✔** Th. 1
Strict core (if non-empty)	✓	**✔** Cor. 1	**✔** Th. 2
Strategy-proof	✓	✓	**✗** Th. 3
Polynomial-time	**✗** Th. 5	✓	**✗** Th. 5

Table 1: Housing market mechanisms: new results are in a bolder font.

We analyzed and compared two recently introduced housing market mechanisms. Whereas it was shown that TTAS may take exponential time, TCR was shown to be strict core selecting just like TTAS. The new and old results are summarized in Table 1. Our abstraction from TTAS and TCR to GATTC helps identify the crucial higher level details and commonality of both TTAS and TCR. This leads to simple proofs for properties satisfied by any GATTC mechanism. Whereas core, strict core, and Pareto optimality are properties that can be fulfilled by any GATTC mechanism, additionally satisfying strategy-proofness requires subtlety in choosing which cycles are implemented in which order. This additional complexity leads to an exponential time lower bound in the case of TTAS and a difficulty in having a very simple description in the case of TCR. It is easily seen that GATTC, and in particular TTAS and TCR not only apply to housing markets but also to other extensions such as agents having multiple number of initial endowments or no endowments or there being some social endowments i.e., not owned initially by any agent.

Our study leads to a number of further research questions. It will be interesting to characterize the subset of GATTC mechanisms which are strategy-proof or are both strategy-proof and polynomial-time. Another question is to see whether being a GATTC mechanism is a necessary condition to simultaneously achieve core stability, Pareto optimality and strict core stability. We have seen that all known housing market mechanisms which are core selecting and Pareto optimal are also strict core selecting (if the strict core is non-empty). This raises the question whether every housing market mechanism which is core selecting and Pareto optimal is also strict core selecting (if the strict core is non-empty).

Acknowledgements

This material is based on work supported by the Deutsche Forschungsgemeinschaft under the grant BR 2312/10-1. The authors thank Jorge Alcalde-Unzu, Felix Brandt, Paul Harrenstein, Elena Molis, Hans Georg Seedig, and the anonymous reviewers of AAAI 2012 and COMSOC 2012 for helpful comments. This paper was previously accepted for presentation at AAAI 2012.

References

J. Alcalde-Unzu and E. Molis. Exchange of indivisible goods and indifferences: The top trading absorbing sets mechanisms. *Games and Economic Behavior*, 73(1):1–16, 2011.

L. Ehlers. Top trading with fixed tie-breaking in markets with indivisible goods. Technical report, March 2012.

P. Jaramillo and V. Manjunath. The difference indifference makes in strategy-proof allocation of objects. Technical Report 1809955, SSRN, 2011.

E. Kalai and D. Schmeidler. An admissible set occurring in various bargaining situations. *Journal of Economic Theory*, 14:402–411, 1977.

J. Ma. Strategy-proofness and the strict core in a market with indivisibilities. *International Journal of Game Theory*, 23(1):75–83, 1994.

T. Quint and J. Wako. On houseswapping, the strict core, segmentation, and linear programming. *Mathematics of Operations Research*, 29(4):861–877, 2004.

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

T. Sönmez and M. U. Ünver. Matching, allocation, and exchange of discrete resources. In J. Benhabib, M. O. Jackson, and A. Bisin, editors, *Handbook of Social Economics*, volume 1 of *Handbooks in Economics*, chapter 17, pages 781–852. Elsevier, 2011.

R. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1 (2):146–160, 1972.

Haris Aziz
Institut für Informatik
Technische Universität München
85748 Garching bei München, Germany
Email: `aziz@in.tum.de`

Bart de Keijzer
Centrum Wiskunde & Informatica (CWI)
1098 XG, Amsterdam, The Netherlands
Email: `B.de.Keijzer@cwi.nl`

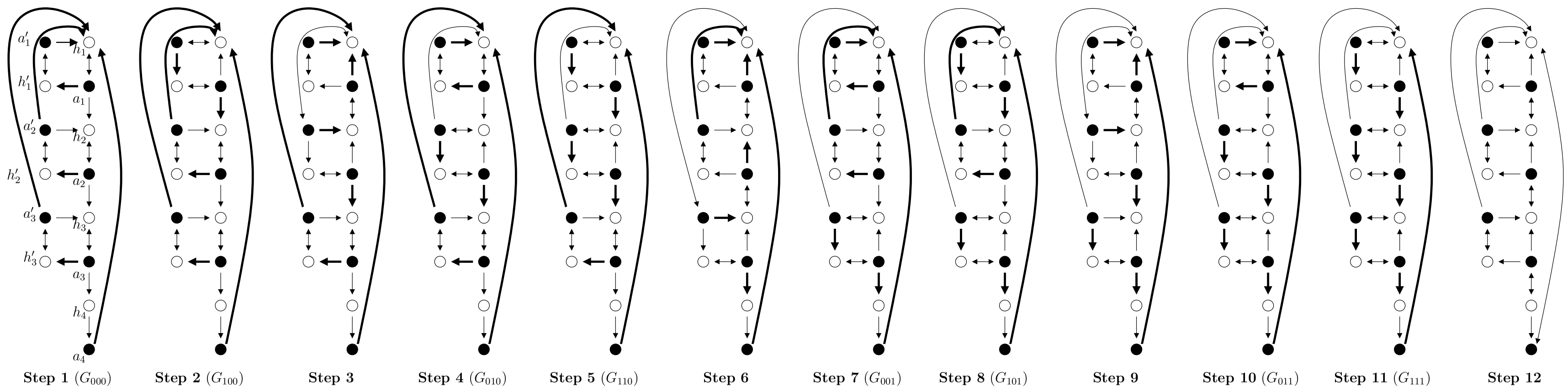

Figure 2: **(Illustrative example for the proof of Theorem 5.)** The graph at the beginning of every step of the TTAS mechanism when it is ran on the instance M_3. Black vertices represent players and white vertices represent houses. When an arc is drawn that has arrows pointing to both its vertices, say vertices a and b, then it stands for the presence of arcs (a, b) and (b, a) in the graph. At the graph for step 1, the names of the vertices are displayed. This is omitted for subsequent steps. In the last step it can be seen that the entire graph is paired symmetric. For every step i except the last one, an arc is displayed in bold in the graph of step i when that arc points from an agent to a house and when that arc is included in the subgraph generated in part 3 of step i (the remaining arcs in this subgraph are all arcs pointing from houses to agents). When in some step, the graph at the beginning of that step equals G_b for some $b \in \{0,1\}^k$, then this is indicated in the figure by the tag "(G_b)" after the step number.