A Simple Family of Top Trading Cycles Mechanisms for Housing Markets with Indifferences

C. Greg Plaxton *

October 26, 2012

Abstract

Recently, important new families of mechanisms have been presented for housing markets with indifferences. These mechanisms are individually rational, strictly Pareto-efficient, strategyproof, and produce an outcome in the strict core when the strict core is nonempty. We propose a novel family of mechanisms and prove that this family achieves the same combination of properties. Our family of mechanisms is based on a generalization of the top trading cycles algorithm. We establish a confluence property of our algorithm, and use this property to give a short proof that the associated mechanisms are strategyproof. We also provide a simple $O(n^3)$-time deterministic implementation of our family of mechanisms, where n denotes the number of agents.

*Department of Computer Science, University of Texas at Austin, 1 University Station C0500, Austin, Texas 78712–0233. Email: plaxton@cs.utexas.edu. This research was supported by NSF Grant CCF–1217980.

1 Introduction

In a seminal paper, Shapley and Scarf [11] study a simple housing market involving n agents. Each agent owns a house, and has a preference order over the n houses. The goal is to determine a suitable allocation of houses to agents, with the understanding that no monetary transfers are allowed. For this setting, Shapley and Scarf [11] present the *top trading cycles (TTC)* algorithm, which they attribute to David Gale. Under weak preferences (i.e., the natural setting in which indifferences are allowed), the TTC algorithm can produce different allocations. Shapley and Scarf show that the set of TTC allocations coincides with the set of competitive allocations, that the set of competitive allocation is contained in the weak core, that the latter containment can be proper (even when preferences are strict), and that the strict core can be empty.

When preferences are strict, the housing market problem is well understood. Roth and Postlewaite [10] show that the strict core corresponds to the unique competitive allocation. Roth [9] shows that the mechanism defined by the TTC algorithm is strategyproof. Bird [4] shows that the TTC mechanism is group strategyproof. Ma [7] shows that the TTC mechanism is the only individually rational, Pareto-efficient, and strategyproof mechanism.

For weak preferences, the housing market prolem is substantially more challenging. Wako [12] shows that the strict core is contained in the set of competitive allocations, and that this containment can be proper. Quint and Wako [8] characterize the class of instances for which the strict core is nonempty, and provide an $O(n^3)$-time algorithm to obtain a strict core allocation on such instances. Recent independent work of Alcalde-Unzu and Molis [2], and of Jaramillo and Manjunath [6], provides new mechanisms for housing markets with weak preferences. These results are highly relevant to the present paper, and are discussed in greater detail below.

Alcalde-Unzu and Molis [2] present an algorithm called Top Trading Absorbing Sets (TTAS). This algorithm yields a family of mechanisms called the TTAS mechanisms. The mechanisms in this family are individually rational, strictly Pareto-efficient, and strategyproof. Furthermore, these mechanisms produce a strict core allocation when the strict core is nonempty. The main shortcoming of the TTAS algorithm is its high time complexity, which arises because TTAS can trade along many "bad" cycles, i.e., cycles where each of the associated agents is already tentatively assigned to a house in its preferred set (amongst the remaining houses). Alcalde-Unzu and Molis leave open the question of whether the TTAS algorithm runs in polynomial time. Aziz and de Keijzer [3] answer this question in the negative by exhibiting a family of instances on which the TTAS algorithm runs in exponential time.

Jaramillo and Manjunath [6] present an algorithm called Top Cycles Rules (TCR). This algorithm yields a family of TCR mechanisms. The mechanisms in this family are individually rational, strictly Pareto-efficient, and strategyproof. Aziz and de Keijzer [3] prove that, like the TTAS family, each mechanism in the TCR family produces an outcome in the strict core when the strict core is nonempty. Thus the TCR mechanisms match the properties mentioned above for the TTAS mechanisms. Furthermore, the TCR algorithm runs in polynomial time. (More specifically, Jaramillo and Manjunath establish an $O(n^6)$ bound on the time complexity of the TCR algorithm.) Like the TTAS algorithm, TCR proceeds iteratively. In each iteration, each remaining agent has a preferred set of houses amongst the remaining houses. Each agent selects a specific house from its preferred set. The cycle trading phase of an iteration then updates the tentative allocation in the usual TTC style, treating the selected houses as the unique top choices. Within this framework, the central question is how an agent should select a specific house from its preferred set.

In the TTAS algorithm, each agent uses a local rule to select a specific house in each iteration. Due to the lack of global coordination, many bad cycles may occur. The TCR algorithm uses an entirely different method to perform the selection. The TCR method (see the description of the "pointing phase" in [6]) is somewhat involved; here we mention only two aspects of this method: (1) in order to achieve a certain "persistence" property, the TCR method takes into account the selections made in the previous iteration (i.e., it is not memoryless); (2) the TCR method involves multiple levels of tie-breakers. These aspects of the TCR method complicate the task of establishing strategyproofness.

We introduce and analyze a new, simpler method for an agent to select a specific house from its preferred set. In each iteration, we compute the shortest path distance of each house to an "unsatisfied" agent, i.e., an agent who is not tentatively assigned to house in its preferred set (see the definition of $distance(G, v)$ in Section 2.1). Each agent selects the house in its preferred set with the smallest distance value, with ties broken according to a fixed total order over the set of houses (see the definition of $next(G, u)$ in Section 2.1). In contrast with the TCR method, our method is memoryless, and uses a single-level tie-breaking scheme. Our analogue of the persistence property of Jaramillo and Manjunath [6] is established as a consequence of the selection method (see Lemmas 3.4, 3.9, and 4.1), and not by complicating the selection method.

In Section 2 we define a nondeterministic algorithm, Algorithm 1. We prove that Algorithm 1 is confluent (see Lemma 4.5), implying that the output does not depend on the nondeterministic choices made during execution (see Theorem 1). In Section 3, we establish properties of the special class of bipartite digraphs that arise in modeling housing markets with indifferences; we use the term "configuration" (formally defined in Section 2.1) to refer to such a structure. In Section 4, we establish properties of the agent preferences, independent of the initial endowments; we use the term "wpp" (formally defined in Section 2.2) to refer to the preferences-related component of a problem instance. In Section 5 we exploit confluence to give a short proof that the family of mechanisms associated with Algorithm 1 is strategyproof (see Lemma 5.1 and Theorem 2). In Appendix E, we again exploit confluence, this time to obtain an $O(n^3)$-time deterministic algorithm, Algorithm 2, with the same input-output behavior as Algorithm 1.

Aziz and de Keijzer [3] introduce a class of mechanisms called Generalized Absorbing Top Trading Cycle (GATTC), which is designed for housing markets with indifferences, and which includes TTC, TTAS, and TCR. The GATTC family is quite broad, and includes mechanisms that are not strategyproof [3]. Aziz and de Keijzer [3] prove that every mechanism in this family is individually rational, strictly Pareto-efficient, and produces an outcome in the strict core if the strict core is nonempty. Since the mechanisms associated with Algorithm 1 are easily seen to belong to the GATTC family, we can focus on establishing strategyproofness. (Remark: Our results do not strongly depend upon the work of Aziz and de Keijzer in the sense that it is straightforward to establish from first principles that our family of mechanisms satisfies individual rationality and strict Pareto-efficiency, and it is also straightforward to use the aforementioned work of Quint and Wako [8] to establish that our family of mechanisms produces an outcome in the strict core if the strict core is nonempty.)

2 A Family of Mechanisms

In Section 1 we have provided an informal description of the family of mechanisms to be analyzed in the present paper. In this section, we provide a formal description. Recall that the main technical challenge of the paper is to establish the strategyproof property of our family of mechanisms. Accordingly, the style of our description is chosen to facilitate formal reasoning.

2.1 Configurations

A *configuration* is a bipartite digraph (U, V, E) where U is a set of agents, V is a totally ordered set of houses, and the following conditions hold: each agent u in U has indegree 1; each house v in V has outdegree 1. (Thus $|U| = |V|$.)

For any configuration $G = (U, V, E)$ and any house v in V, we define $agent(G, v)$ as the unique agent u such that edge (v, u) belongs to E. For any configuration $G = (U, V, E)$ and any agent u in U, we define $house(G, u)$ as the unique house v such that $agent(G, v) = u$.

For any configuration $G = (U, V, E)$, we define $allocation(G)$ as the allocation that assigns each house v in V to $agent(G, v)$.

For any configuration $G = (U, V, E)$, and any agent u in U, we define $\Gamma(G, u)$ as $\{v \mid (u, v) \in E\}$. A configuration $G = (U, V, E)$ is *initial* if $\Gamma(G, u)$ is empty for all agents u in U.

For any configuration $G = (U, V, E)$, we define $\mathit{satisfied}(G)$ as the set of all agents u in U such that $house(G, u)$ belongs to $\Gamma(G, u)$, and we define $\mathit{unsatisfied}(G)$ as $U \setminus \mathit{satisfied}(G)$. A configuration G is *final* if $\mathit{unsatisfied}(G)$ is empty.

For any configuration $G = (U, V, E)$ and any house v in V, we define $distance(G, v)$ as the length of a shortest path from v to an agent in $\mathit{unsatisfied}(G)$. If there is no such path, we define $distance(G, v)$ as ∞.

For any configuration $G = (U, V, E)$ and any agent u in U, we define $next(G, u)$ as follows: if $distance(G, v) = \infty$ for all v in $\Gamma(G, u)$, then $next(G, u) = nil$; otherwise, letting V' denote the set of all v in $\Gamma(G, u)$ minimizing $distance(G, v)$, we define $next(G, u)$ as the minimum element of V'.

For any configuration $G = (U, V, E)$, we define $pruned(G)$ as the configuration $G' = (U, V, E \setminus E')$ where E' denotes

$$\{(u, v) \in E \mid u \in U \ \wedge \ v \neq next(G, u)\},$$

and we define $cycles(G)$ as the set of all directed cycles in $pruned(G)$.

For any configuration $G = (U, V, E)$ and any cycle C in $cycles(G)$, we define $trade(G, C)$ as the configuration $(U, V, (E \setminus E') \cup E'')$ where E' denotes

$$\{(v, u) \in V \times U \mid v \in C \ \wedge \ agent(G, v) = u\}$$

and E'' denotes

$$\{(v, u) \in V \times U \mid u \in C \ \wedge \ next(G, u) = v\}.$$

For any configuration G, we define $exhausted(G)$ as the set of all agents u in $\mathit{unsatisfied}(G)$ such that $next(G, u) = nil$.

2.2 Preferences

A *weak preference relation* is a total preorder (also known as a weak order). We define a *weak preference profile (wpp)* as a triple $(U, V, \succsim)$ where U is a set of agents, V is a totally ordered set of houses such that $|U| = |V|$, and $\succsim$ is a function from U to the set of weak preference relations over V. Notation: Given a wpp $(U, V, \succsim)$, and an agent u in U, we write $\succsim_u$ to refer to the weak preference relation to which u is mapped by $\succsim$; we use the symbol $\sim$ to denote indifference and the symbol $\succ$ to denote strict preference.

For any wpp $W = (U, V, \succsim)$, we define $\mathit{configs}(W)$ as the set of all configurations $G = (U, V, E)$ such that for any agent u in U, any house v in $\Gamma(G, u)$, and any house v' in $V \setminus \Gamma(G, u)$, we have $v \succ_u v'$.

Lemma 2.1. *For any wpp $W = (U, V, \succsim)$, any configuration $G = (U, V, E)$ in $\mathit{configs}(W)$, and any cycle C in $\mathit{cycles}(G)$, the configuration $\mathit{trade}(G, C)$ belongs to $\mathit{configs}(W)$.*

Proof. Straightforward. □

For any wpp $W = (U, V, \succsim)$, any agent u in U, and any subset V' of V, we define $\mathit{top}(W, u, V')$ as the set of houses v in V' such that $v \succsim_u v'$ holds for all v' in V'.

For any wpp $W = (U, V, \succsim)$, any configuration $G = (U, V, E)$ in $\mathit{configs}(W)$, and any agent u in $\mathit{exhausted}(G)$, we define $\mathit{reveal}(W, G, u)$ as the configuration $(U, V, E \cup E')$ where

$$E' = \{(u, v) \mid v \in \mathit{top}(W, u, V \setminus \Gamma(G, u))\}.$$

Lemma 2.2. *For any wpp $W = (U, V, \succsim)$, any configuration $G = (U, V, E)$ in $\mathit{configs}(W)$, and any agent u in $\mathit{exhausted}(G)$, the configuration $\mathit{reveal}(W, G, u)$ belongs to $\mathit{configs}(W)$.*

Proof. Straightforward. □

For any wpp $W = (U, V, \succsim)$, let $\mathit{moves}(W)$ denote the edge-labeled digraph with vertex set $\mathit{configs}(W)$ and edge set determined as follows. First, for any G in $\mathit{configs}(W)$ and any C in $\mathit{cycles}(G)$, there is an edge (G, G') with label C, where $G' = \mathit{trade}(G, C)$ belongs to $\mathit{configs}(W)$ by Lemma 2.1. Second, for any G in $\mathit{configs}(W)$ and any agent u in $\mathit{exhausted}(G)$, there is an edge (G, G') with label u, where $G' = \mathit{reveal}(W, G, u)$ belongs to $\mathit{configs}(W)$ by Lemma 2.2.

For any wpp W and any G in $\mathit{configs}(W)$, we define $\Gamma(W, G)$ as the set of all configurations G' such that edge (G, G') belongs to $\mathit{moves}(W)$.

2.3 A Nondeterministic Algorithm

A *housing market instance* is a pair (W, G) where W is a wpp and G is an initial configuration in $\mathit{configs}(W)$.

We will first study the simple nondeterministic algorithm of Figure 1, which we refer to as Algorithm 1. We show that Algorithm 1 terminates with a configuration G that is final (Lemma 4.3). We also show that the output allocation is uniquely determined (Theorem 1). In Appendix E, we present a simple deterministic algorithm for computing the output allocation in $O(n^3)$ time.

Since Algorithm 1 assumes a total ordering over the set of houses (see the tie-breaker in the definition of $\mathit{next}(G, u)$ in Section 2.1), it defines a family of house allocation mechanisms, as

while $\Gamma(W, G) \neq \emptyset$
    $G :=$ a nondeterministically chosen element of $\Gamma(W, G)$
return $allocation(G)$

Figure 1: We refer to the above nondeterministic algorithm as Algorithm 1. Initially, (W, G) is a housing market instance.

opposed to a single mechanism. The related works [2, 3, 6] discussed in Section 1 share the same characteristic.

Remark: In this paper we assume that we are given a total ordering over the set of houses. If instead we are given a total ordering over the set of agents, then we can use this ordering, together with the initial allocation, to induce a total ordering over the set of houses.

3 Properties of Configurations

Lemma 3.1. *Let $G = (U, V, E)$ be a configuration and let $G' = pruned(G)$. Then $distance(G', v) = distance(G, v)$ for all houses v in V, and $next(G', u) = next(G, u)$ for all agents u in U.*

Proof. Let $P(i)$ denote the predicate "for any house v in V such that $distance(G, v) = 2i + 1$, we have $distance(G', v) \leq 2i + 1$". We use induction to prove that $P(i)$ holds for all nonnegative integers i.

Base case: $i = 0$. Let v be a house such that $distance(G, v) = 1$. Let u denote $agent(G, v)$. Thus u belongs to $unsatisfied(G)$, and hence also belongs to $unsatisfied(G')$. Since G' includes the edge (v, u), we conclude that $distance(G', v) = 1$. Hence $P(0)$ holds.

Induction step. Let i be a nonnegative integer, and assume that $P(i)$ holds. Let v be a house such that $distance(G, v) = 2i + 3$. Let u denote $agent(G, v)$. Let v' denote $next(G, u)$. Thus $distance(G, v') = 2i + 1$, and the induction hypothesis implies that $distance(G', v') \leq 2i + 1$. Since the edges (v, u) and (u, v') belong to $pruned(G')$, we conclude that $distance(G', v) \leq 2i+3$, as required. Hence $P(i + 1)$ holds, completing our proof by induction.

Since $P(i)$ holds for all nonnegative integers i, and since $distance(G, v) = \infty$ implies $distance(G', v) \leq distance(G, v)$, we conclude that $distance(G', v) \leq distance(G, v)$ for all houses v in V. On the other hand, since G' is a subgraph of G, $distance(G', v) \geq distance(G, v)$ for all houses v in V. Thus $distance(G', v) = distance(G, v)$ for all houses v in V.

Let u be an agent in U. We prove that $next(G', u) = next(G, u)$ by considering two cases.

Case 1: $next(G, u) = nil$. Thus $\Gamma(G', u)$ is empty, and hence $next(G', u) = nil$.

Case 2: $next(G, u) \neq nil$. Let v denote $next(G, u)$. Thus $distance(G, v)$ is finite. Since $distance(G', v) = distance(G, v)$, we conclude that $distance(G', v)$ is finite. Since $\Gamma(G', u) = \{v\}$ and $distance(G', v)$ is finite, we have $next(G', u) = v$, as required. □

Lemma 3.2. *Let G be a configuration, and let C belong to $cycles(G)$. Then at least one agent in $unsatisfied(G)$ is on C.*

Proof. Immediate from Lemma 3.1. □

Lemma 3.3. *Let G be a configuration, let C belong to $cycles(G)$, and let u be an agent on C. Then u belongs to $satisfied(trade(G, C))$.*

Proof. Immediate from the definition of $trade(G, C)$. □

Lemma 3.4. *Let $G = (U, V, E)$ be a configuration, let C belong to $cycles(G)$, let G' denote $trade(G, C)$, and let v belong to V. Then $distance(G', v) \geq distance(G, v)$.*

Proof. Immediate from Lemma A.1, which is proven in Appendix A. □

For any configuration G, we define $frozen(G)$ as the set of all agents u in $satisfied(G)$ such that $next(G, u) = nil$.

Lemma 3.5. *Let $G = (U, V, E)$ be a configuration, let u belong to U, and let v denote $house(G, u)$. Then $distance(G, v) = \infty$ if and only if u belongs to $frozen(G)$.*

Proof. If u belongs to $frozen(G)$, then $distance(G, v) = \infty$. Assume that $distance(G, v) = \infty$. We consider two cases.

Case 1: u belongs to $satisfied(G)$. Since $distance(G, v) = \infty$, we deduce that $next(G, u) = nil$. Hence u belongs to $frozen(G)$, as required.

Case 2: u belongs to $unsatisfied(G)$. Then $distance(G, v) = 1$, a contradiction. □

Lemma 3.6. *Let G be a configuration. Then G is final if and only if $cycles(G)$ and $exhausted(G)$ are empty.*

Proof. For the "only if" direction, assume that G is final. Hence $unsatisfied(G)$ is empty. Thus Lemma 3.2 implies that $cycles(G)$ is empty, and the definition of $exhausted(G)$ implies that $exhausted(G)$ is empty.

For the "if" direction, assume that $cycles(G)$ and $exhausted(G)$ are empty. Let G be of the form (U, V, E) and let U' denote $U \setminus frozen(G)$. Since U' is disjoint from $exhausted(G) \cup frozen(G)$, we deduce that $next(G, u) \neq nil$ for all agents u in U'. Thus there is a cycle in $pruned(G)$ unless U' is empty. Since $cycles(G)$ is empty, we conclude that U' is empty, and hence $frozen(G) = U$. Thus $unsatisfied(G)$ is empty, and G is final. □

For any configurations $G = (U, V, E)$ and $G' = (U', V', E')$, we write $G \lesssim G'$ to mean that the following conditions are satisfied: $U = U'$; $V = V'$; for all agents u in U, $\Gamma(G', u)$ contains $\Gamma(G, u)$; for all agents u in $satisfied(G)$, $\Gamma(G', u) = \Gamma(G, u)$; $satisfied(G')$ contains $satisfied(G)$; for all houses v in V, $distance(G', v) \geq distance(G, v)$; for all agents u in $frozen(G) \cup unsatisfied(G')$, $house(G', u) = house(G, u)$. Thus for any configuration G, we have $G \lesssim G$. Lemma 3.8 below establishes that the relation $\lesssim$ defines a preorder over the set of all configurations.

Lemma 3.7. *Let G and G' be configurations such that $G \lesssim G'$. Then $frozen(G')$ contains $frozen(G)$.*

Proof. Let u be an agent in $frozen(G)$ and let v denote $house(G, u)$. Since $G \lesssim G'$, we have $distance(G', v) \geq distance(G, v)$. Since Lemma 3.5 implies that $distance(G, v) = \infty$, we deduce that $distance(G', v) = \infty$. Hence Lemma 3.5 implies that u belongs to $frozen(G')$. □

Lemma 3.8. *Let G, G', and G'' be configurations such that $G \lesssim G'$ and $G' \lesssim G''$. Then $G \lesssim G''$.*

Proof. Immediate from the definition of $\lesssim$ and Lemma 3.7. $\square$

Lemma 3.9. *Let $G = (U, V, E)$ be a configuration, let u be an agent such that $next(G, u) \neq nil$, let v denote $next(G, u)$, and let G' be a configuration such that $G \lesssim G'$, $distance(G', v) = distance(G, v)$, and $\Gamma(G', u) = \Gamma(G, u)$. Then $next(G', u) = next(G, u)$.*

Proof. Since $G \lesssim G'$, we have $distance(G', v') \geq distance(G, v')$ for all v' in V. Since $\Gamma(G', u) = \Gamma(G, u)$ and $distance(G', v) = distance(G, v)$, we conclude that $next(G', u) = next(G, u)$, as required. $\square$

Lemma 3.10. *Let $G = (U, V, E)$ be a configuration, let C belong to $cycles(G)$, and let G' denote $trade(G, C)$. Then $G \lesssim G'$ holds.*

Proof. The configuration G' is of the form (U, V, E'), and for all agents u in U, we have $\Gamma(G', u) = \Gamma(G, u)$. Lemmas 3.2 and 3.3 imply that $satisfied(G')$ properly contains $satisfied(G)$. Lemma 3.4 implies that $distance(G', v) \geq distance(G, v)$ for all v in V. Since $next(G, u) = nil$ for any agent u in $frozen(G)$, no agent on C belongs to $frozen(G)$. Hence for all agents u in $frozen(G)$, $house(G', u) = house(G, u)$. Lemma 3.3 implies that all agents u on C belong to $satisfied(G')$. Thus for all agents u in $unsatisfied(G')$, we have $house(G', u) = house(G, u)$. The claim of the lemma follows. $\square$

4 Properties of Wpps

Lemma 4.1. *Let $W = (U, V, \succsim)$ be a wpp, let G belong to $configs(W)$, let u belong to $exhausted(G)$, let G' denote $reveal(W, G, u)$, and let u' belong to $U - u$. Then $G \lesssim G'$. Furthermore, if u belongs to $unsatisfied(G')$ or u' belongs to a cycle in $cycles(G)$, then $next(G', u') = next(G, u')$.*

Proof. The claim that $G \lesssim G'$ holds is immediate from Lemma B.2, which is proven in Appendix B.

Assume that u belongs to $unsatisfied(G')$. Hence Lemma B.2 implies that $distance(G', v) = distance(G, v)$ for all houses v in V. Since $\Gamma(G', u')$ is equal to $\Gamma(G, u')$, we deduce that $next(G', u')$ is equal to $next(G, u')$, as required.

Assume that u' belongs to a cycle C in $cycles(G)$. Thus $next(G, u')$ is a house on C; let v denote $next(G, u')$. Lemma 3.2 implies that $attractor(G, v)$ (see Appendix B for the definition of $attractor(G, v)$) is an agent on C, and hence is not equal to u. Thus Lemma B.2 implies $distance(G', v) = distance(G, v)$. Since $G \lesssim G'$, $distance(G', v) = distance(G, v)$, and $\Gamma(G', u') = \Gamma(G, u')$, Lemma 3.9 implies that $next(G', u') = next(G, u')$, as required. $\square$

For any wpp $W = (U, V, \succsim)$, any subset U' of U, and any G in $configs(W)$, we define $\Gamma(W, G, U')$ as the set of all configurations G' such that (G, G') is an edge in $moves(W)$ and the label of edge (G, G') is neither an agent in U' nor a cycle that includes an agent in U', and we define $\Gamma^*(W, G, U')$ as the set of all configurations G' in $configs(W)$ for which there exists a nonnegative integer k and a sequence of configurations G_i, $0 \leq i \leq k$, such that the following conditions hold: $G_0 = G$; $G_k = G'$; G_{i+1} belongs to $\Gamma(W, G_i, U')$ for $0 \leq i < k$.

For any wpp W, and any configuration G in $configs(W)$, we define $\Gamma^*(W, G)$ as $\Gamma^*(W, G, \emptyset)$.

Lemma 4.2. *Let W be a wpp, let G belong to $\mathit{configs}(W)$, and let G' belong to $\Gamma^*(W, G)$. Then $G \lesssim G'$.*

Proof. Immediate from Lemmas 3.8, 3.10, and 4.1. □

The next lemma shows that the nondeterministic algorithm of Figure 1 terminates within a polynomial number of iterations. We present a faster implementation in Appendix E.

Lemma 4.3. *Consider an execution of the while loop of Algorithm 1 on a wpp $W = (U, V, \succsim)$ and an initial configuration G in $\mathit{configs}(W)$. Then the while loop terminates with a final configuration within at most $|V|^2 + |V|$ iterations.*

Proof. For any configuration $G' = (U, V, E)$, let $f(G')$ denote $|\mathit{satisfied}(G')|$, and let $g(G')$ denote $|E| - |V|$. Thus $0 \leq f(G') \leq |V|$ and $0 \leq g(G') \leq |V|^2$. Let G_i denote the configuration associated with the variable G after i iterations of the while loop have been completed. Thus $f(G_0) = g(G_0) = 0$. Lemma 4.2 implies that $f(G_{i+1}) \geq f(G_i)$ and $g(G_{i+1}) \geq g(G_i)$ for all $i \geq 0$. Lemmas 3.2, 3.3, and 4.2 together imply that if $G_{i+1} = \mathit{trade}(G_i, C)$ for some C in $\mathit{cycles}(G_i)$, then $f(G_{i+1}) \geq f(G_i) + 1$; thus we can have at most $|V|$ iterations in this category. If $G_{i+1} = \mathit{reveal}(W, G_i, u)$ for some u in $\mathit{exhausted}(G_i)$, then $g(G_{i+1}) \geq g(G_i) + 1$; thus we can have at most $|V|^2$ iterations in this category.

Lemma 3.6 implies that the configuration corresponding to program variable G is final when the while loop of Algorithm 1 terminates. □

Lemma 4.4. *Let $W = (U, V, \succsim)$ be a wpp, let G be a configuration in $\mathit{configs}(W)$, let G' belong to $\Gamma(W, G)$, let ℓ be the label of edge (G, G') in $\mathit{moves}(W)$, and let G'' be a configuration in $\Gamma^*(W, G)$ such that there is no label-ℓ edge outgoing from G''. Then G'' belongs to $\Gamma^*(W, G')$.*

Proof. Immediate from Lemmas C.1, C.2, C.3, C.4, C.5, and C.6, which are proven in Appendix C. □

For any wpp $W = (U, V, \succsim)$, any subset U' of U, and any configuration G in $\mathit{configs}(W)$, we define $\mathit{sinks}(W, G, U')$ as the set of all configurations G' in $\Gamma^*(W, G, U')$ such that $\Gamma(W, G', U')$ is empty.

Lemma 4.5. *Let $W = (U, V, \succsim)$ be a wpp, let U' be a subset of U, and let G belong to $\mathit{configs}(W)$. Then $|\mathit{sinks}(W, G, U')| = 1$.*

Proof. Lemma 4.3 implies that $\mathit{sinks}(W, G, U')$ is nonempty. Let G^* belong to $\mathit{sinks}(W, G, U')$. Thus G^* is contained in $\Gamma^*(W, G, U')$ and $\mathit{sinks}(W, G^*, U') = \{G^*\}$.

Lemma 4.4 implies that $\mathit{sinks}(W, G, U') = \mathit{sinks}(W, G', U')$ for all G' in $\Gamma(W, G, U')$. Thus, by repeated application of Lemma 4.4, we deduce that $\mathit{sinks}(W, G, U') = \mathit{sinks}(W, G', U')$ for all G' in $\Gamma^*(W, G, U')$. Since G^* is contained in $\Gamma^*(W, G, U')$, we find that $\mathit{sinks}(W, G, U') = \mathit{sinks}(W, G^*, U') = \{G^*\}$. The claim of the lemma follows. □

For any wpp $W = (U, V, \succsim)$, any configuration G in $\mathit{configs}(W)$, any agent u in U, and any house v in V, we define the predicate $\mathit{bottom}(W, G, u, v)$ to mean that v belongs to $\Gamma(G, u)$ and $v' \succsim_u v$ holds for all houses v' in $\Gamma(G, u)$.

For any wpp $W = (U, V, \succsim)$, we define $\mathit{admissible}(W)$ as the set of all configurations G in $\mathit{configs}(W)$ such that the following conditions hold: for any agent u in $\mathit{frozen}(G)$, we have $\mathit{bottom}(W, G, u, \mathit{house}(G, u))$; for any agent u in $U \setminus \mathit{frozen}(G)$ and any house v in $\Gamma(G, u)$ such that $\mathit{agent}(G, v)$ does not belong to $\mathit{frozen}(G)$, we have $\mathit{bottom}(W, G, u, v)$.

Lemma 4.6. *Let $W = (U, V, \succsim)$ be a wpp, let G belong to $\mathit{admissible}(W)$, let G' belong to $\Gamma^*(W, G)$, let u belong to U, let v denote $\mathit{house}(G, u)$, and let v' denote $\mathit{house}(G', u)$. Then G' belongs to $\mathit{admissible}(W)$ and $v' \succsim_u v$. Furthermore, if u belongs to $\mathit{satisfied}(G)$ then $v' \sim_u v$.*

Proof. By induction using Lemmas D.1 and D.2, which are proven in Appendix D. □

Lemma 4.7. *Let W be a wpp and let G be an initial configuration in $\mathit{configs}(W)$. Then G belongs to $\mathit{admissible}(W)$.*

Proof. Straightforward. □

For any wpp $W = (U, V, \succsim)$, any G in $\mathit{configs}(W)$, and any subset U' of U, we define $\mathit{sink}(W, G, U')$ as the unique (by Lemma 4.5) element of $\mathit{sinks}(W, G, U')$.

For any wpp W and any G in $\mathit{configs}(W)$, we define $\mathit{sink}(W, G)$ as $\mathit{sink}(W, G, \emptyset)$.

For any wpp $W = (U, V, \succsim)$ and any agent u in U, we define $\mathit{sinks}(W, u)$ as the set of all configurations G in $\mathit{admissible}(W)$ such that $\Gamma(W, G, \{u\})$ is empty and u belongs to $\mathit{unsatisfied}(G)$.

For any configuration G and any agent u in $\mathit{unsatisfied}(G)$, we define $\mathit{reach}(G, u)$ as the set of all houses v such that there is a path from v to u in $\mathit{pruned}(G)$.

Lemma 4.8. *Let $W = (U, V, \succsim)$ be a wpp, let u belong to U, let G be a configuration in $\mathit{sinks}(W, u)$, and let v belong to V. Then exactly one of the following two conditions is satisfied: $\mathit{agent}(G, v)$ belongs to $\mathit{frozen}(G)$; v belongs to $\mathit{reach}(G, u)$.*

Proof. Let u^* denote $\mathit{agent}(G, v)$. We consider two cases.

Case 1: u^* belongs to $\mathit{frozen}(G)$. Thus u^* belongs to $\mathit{satisfied}(G)$ and $\mathit{next}(G, u^*) = \mathit{nil}$. Since u^* belongs to $\mathit{satisfied}(G)$ and G belongs to $\mathit{sinks}(W, u)$, we deduce that $u^* \neq u$. Since $\mathit{next}(G, u^*) = \mathit{nil}$, agent u^* is the only agent reachable via a path from v in $\mathit{pruned}(G)$. Since $u \neq u^*$, we conclude that v does not belong to $\mathit{reach}(G, u)$.

Case 2: u^* does not belong to $\mathit{frozen}(G)$. Consider the path P in $\mathit{pruned}(G)$ obtained by starting at v and repeatedly following the outgoing edge of the current vertex until (1) a cycle C is formed or (2) an agent u' is reached such that $\mathit{next}(G, u') = \mathit{nil}$. If (1) occurs, then since G belongs to $\mathit{sinks}(W, u)$, we deduce that u belongs to C and hence that v belongs to $\mathit{reach}(G, u)$.

Now assume that (2) occurs, and let v' denote $\mathit{house}(G, u')$. We claim that $\mathit{distance}(G, v')$ is finite. If $v = v'$, the claim follows from Lemma 3.5 and the Case 2 condition. Assume $v \neq v'$ and let u'' denote the agent preceding v' on P. Then $v' = \mathit{next}(G, u'')$ and hence $\mathit{distance}(G, v')$ is finite, completing the proof of the claim. The claim, together with Lemma 3.5, implies that u' does not belong to $\mathit{frozen}(G)$. Since $\mathit{next}(G, u') = \mathit{nil}$ and u' does not belong to $\mathit{frozen}(G)$, we deduce that u' belongs to $\mathit{unsatisfied}(G)$ and hence that u' belongs to $\mathit{exhausted}(G)$. Since G belongs to $\mathit{sinks}(W, u)$ and u' belongs to $\mathit{exhausted}(G)$, we deduce that $u' = u$ and hence that v belongs to $\mathit{reach}(G, u)$. □

Lemma 4.9. *Let $W = (U, V, \succsim)$ be a wpp, let u belong to U, and let G be a configuration in $\mathit{sinks}(W, u)$. Then $|\Gamma(W, G)| = 1$ and the following claims hold, where G' denotes the unique configuration in $\Gamma(W, G)$.*

1. *If u belongs to $\mathit{unsatisfied}(G')$, then $\mathit{reach}(G', u) = \mathit{reach}(G, u)$ and G' belongs to $\mathit{sinks}(W, u)$.*

2. *If u belongs to $\mathit{satisfied}(G')$, then $\mathit{house}(G', u)$ belongs to $\mathit{top}(W, u, \mathit{reach}(G, u))$.*

Proof. We first argue that $|\Gamma(W, G)| = 1$. Since G belongs to $\mathit{sinks}(W, u)$, we find that $\Gamma(W, G, \{u\})$ is empty. Hence u appears on any cycle C in $\mathit{cycles}(G)$. Since the cycles in $\mathit{cycles}(G)$ are disjoint, we have $|\mathit{cycles}(G)| \leq 1$. We consider two cases.

Case 1: $\mathit{next}(G, u) = \mathit{nil}$. Since u belongs to $\mathit{unsatisfied}(G)$, we deduce that u belongs to $\mathit{exhausted}(G)$ and that $\mathit{cycles}(G)$ is empty. It follows that $\Gamma(W, G) = \{\mathit{reveal}(W, G, u)\}$. Hence $|\Gamma(W, G)| = 1$.

Case 2: $\mathit{next}(G, u) \neq \mathit{nil}$. Thus u does not belong to $\mathit{exhausted}(G)$, and hence $\mathit{exhausted}(G)$ is empty. Since u belongs to $\mathit{unsatisfied}(G)$, Lemma 3.6 implies that $\mathit{cycles}(G)$ is nonempty. Since $|\mathit{cycles}(G)| \leq 1$, we deduce that $|\mathit{cycles}(G)| = 1$ and hence $|\Gamma(W, G)| = 1$.

We now address Claim 1. Assume that u belongs to $\mathit{unsatisfied}(G')$. Thus $G' = \mathit{reveal}(W, G, u)$ and u belongs to $\mathit{exhausted}(G)$. Hence Lemma 4.1 implies that $G \lesssim G'$ and $\mathit{next}(G', u')$ is equal to $\mathit{next}(G, u')$ for all agents u' in $U - u$. It follows that $\mathit{reach}(G', u)$ is equal to $\mathit{reach}(G, u)$ and $\mathit{exhausted}(G') \cap (U - u)$ is equal to $\mathit{exhausted}(G) \cap (U - u)$. Since G belongs to $\mathit{sinks}(W, u)$, we find that $\mathit{exhausted}(G) \cap (U - u)$ is empty. Thus $\mathit{exhausted}(G') \cap (U - u)$ is empty. Furthermore, if C belong to $\mathit{cycles}(G')$ and u is not on C, then C belongs to $\mathit{cycles}(G)$; since G belongs to $\mathit{sinks}(W, u)$, we deduce that u belongs to every cycle in $\mathit{cycles}(G')$. Since $\mathit{exhausted}(G') \cap (U - u)$ is empty, we conclude that $\Gamma(W, G', \{u\})$ is empty.

Since G belongs to $\mathit{sinks}(W, u)$, we know that G belongs to $\mathit{admissible}(W)$. Hence Lemma 4.6 (or simply Lemma D.1) implies that G' belongs to $\mathit{admissible}(W)$.

Since G' is a configuration in $\mathit{admissible}(W)$ such that $\Gamma(W, G', \{u\})$ is empty and u belongs to $\mathit{unsatisfied}(G')$, we conclude that G' belongs to $\mathit{sinks}(W, u)$.

It remains to address Claim 2. Assume that u belongs to $\mathit{satisfied}(G')$. Let v denote $\mathit{house}(G, u)$, let v' denote $\mathit{house}(G', u)$, and let u' denote $\mathit{agent}(G, v')$. Let V_0 denote the set of all houses v_0 such that $\mathit{agent}(G, v_0)$ belongs to $\mathit{frozen}(G)$. Lemma 4.2 implies $G \lesssim G'$, and hence that v' does not belong to V_0. By Lemma 4.8, $V_0 = V \setminus \mathit{reach}(G, u)$. Thus v' belongs to $\mathit{reach}(G, u)$. We consider two cases.

Case 1: u does not belong to $\mathit{exhausted}(G)$. Thus $G' = \mathit{trade}(G, C)$ for some C in $\mathit{cycles}(G)$ such that u is on C, $\Gamma(G, u) = \Gamma(G', u)$, and v' belongs to $\Gamma(G, u)$. Let V_1 denote $\mathit{reach}(G, u) \cap \Gamma(G, u)$, and let V_2 denote $\mathit{reach}(G, u) \setminus V_1$. Thus v' belongs to V_1. Since G belongs to $\mathit{admissible}(W)$, we have $v' \sim_u v''$ for all houses v'' in V_1. Since G belongs to $\mathit{configs}(W)$, we have $v' \succ_u v''$ for all houses v'' in V_2. We conclude that v' belongs to $\mathit{top}(W, u, V_1 \cup V_2) = \mathit{top}(W, u, \mathit{reach}(G, u))$, as required.

Case 2: u belongs to $\mathit{exhausted}(G)$. Thus $G' = \mathit{reveal}(W, G, u)$, $v' = v$, and it follows that $v' \succsim_u v''$ holds for all houses v'' in $V \setminus V_0 = \mathit{reach}(G, u)$. Thus v' belongs to $\mathit{top}(W, u, \mathit{reach}(G, u))$, as required. □

Lemma 4.10. *Let $W = (U, V, \succsim)$ be a wpp, let u belong to U, and let G belong to $\mathit{sinks}(W, u)$. Then $\mathit{house}(\mathit{sink}(W, G), u)$ belongs to $\mathit{top}(W, u, \mathit{reach}(G, u))$.*

Proof. By repeated application of Lemma 4.9. □

5 Main Results

For any housing market instance $I = (W, G)$, we define $\mathit{sink}(I)$ as $\mathit{sink}(W, G)$. Theorem 1 below establishes that Algorithm 1 defines a deterministic mechanism.

Theorem 1. *Let I be a housing market instance. Then any execution of Algorithm 1 on instance I produces the same allocation.*

Proof. Any execution of Algorithm 1 on instance I returns $\mathit{allocation}(\mathit{sink}(I))$. $\square$

For any housing market instance $I = (W, G)$ where $W = (U, V, \succsim)$ and any agent u in U, we define $\mathit{sink}(I, u)$ as $\mathit{sink}(W, G, \{u\})$, we define $\mathit{houses}(I, u)$ as $\mathit{reach}(\mathit{sink}(I, u), u)$, and we define $\mathit{house}(I, u)$ as $\mathit{house}(\mathit{sink}(I), u)$.

Lemma 5.1. *Let $I = (W, G)$ be a housing market instance with $W = (U, V, \succsim)$, and let u belong to U. Then $\mathit{house}(I, u)$ belongs to $\mathit{top}(W, u, \mathit{houses}(I, u))$.*

Proof. Let G' denote $\mathit{sink}(I, u)$. Thus G' belongs to $\Gamma^*(W, G, \{u\})$ and $\Gamma(W, G', \{u\})$ is empty. Since Lemma 4.7 implies that G belongs to $\mathit{admissible}(W)$, Lemma 4.6 implies that G' belongs to $\mathit{admissible}(W)$. Since I is a housing market instance, the configuration G is initial, and hence $\Gamma(G, u)$ is empty. Since $\Gamma(G, u)$ is empty and G' belongs to $\Gamma^*(W, G, \{u\})$, we find that $\Gamma(G', u)$ is empty and hence u belongs to $\mathit{unsatisfied}(G')$. Since G' belongs to $\mathit{admissible}(W)$, $\Gamma(W, G', \{u\})$ is empty, and u belongs to $\mathit{unsatisfied}(G')$, we conclude that G' belongs to $\mathit{sinks}(W, u)$. Thus Lemma 4.10 implies that $\mathit{house}(\mathit{sink}(W, G'), u)$ belongs to $\mathit{top}(W, u, \mathit{reach}(G', u))$. Theorem 1 implies that $\mathit{sink}(I) = \mathit{sink}(W, G')$. Hence $\mathit{house}(I, u)$ belongs to $\mathit{top}(W, u, \mathit{reach}(G', u))$. By definition, $\mathit{houses}(I, u)$ is equal to $\mathit{reach}(G', u)$, and the claim of the lemma follows. $\square$

For any wpps $W = (U, V, \succsim)$ and $W' = (U, V, \succsim')$, and any u in U, we define the predicate $\mathit{equiv}(W, W', u)$ to hold if $v \succsim_{u'} v'$ is logically equivalent to $v \succsim'_{u'} v'$ for all agents u' in $U - u$ and all houses v and v' in V.

Lemma 5.2. *Let $G = (U, V, E)$ be an initial configuration, let u belong to U, and let $I = (W, G)$ and $I' = (W', G)$ be housing market instances such that $\mathit{equiv}(W, W', u)$ holds. Then $\mathit{sink}(I, u) = \mathit{sink}(I', u)$ and $\mathit{houses}(I, u) = \mathit{houses}(I', u)$.*

Proof. The first equation holds because $\mathit{sink}(I, u)$ and $\mathit{sink}(I', u)$ are each independent of the preferences of u. The second equation follows since $\mathit{houses}(I, u) = \mathit{reach}(\mathit{sink}(I, u), u) = \mathit{reach}(\mathit{sink}(I', u), u) = \mathit{houses}(I', u)$. $\square$

Theorem 2. *Each mechanism in the family associated with Algorithm 1 is individually rational, strictly Pareto-efficient, strategyproof, and produces an outcome in the strict core whenever the strict core is nonempty.*

Proof. As discussed at the end of Section 1, the family of mechanisms associated with Algorithm 1 lies within the broad class of GATTC mechanisms introduced by Aziz and de Keijzer [3]. Every mechanism in the GATTC family is individually rational, strictly Pareto-efficient, and produces and outcome in the strict core whenever the strict core is nonempty [3]. It remains only to establish strategyproofness.

Let $I = (W, G)$ be a housing market instance with $W = (U, V, \succsim)$, and let u be an agent in U such that $\succsim$ reflects u's true preferences over the houses in V. Let $I' = (W', G)$ be a housing market instance such that $\mathit{equiv}(W, W', u)$ holds. Let V' denote $\mathit{houses}(I, u)$, which is equal to $\mathit{houses}(I', u)$ by Lemma 5.2. Lemma 5.1 implies that $\mathit{house}(I, u)$ belongs to $\mathit{top}(W, u, V')$ and that $\mathit{house}(I', u)$ belongs to V'. Hence $\mathit{house}(I, u) \succsim_u \mathit{house}(I', u)$. $\square$

6 Concluding Remarks

Abraham et al. [1] show how to generalize the $O(n^{2.5})$-time Hopcroft-Karp algorithm [5] for computing a maximum cardinality matching in a bipartite graph to obtain the same asymptotic time bound for computing maximum cardinality Pareto-optimal matchings for house allocation problems. For the setting with indifferences considered in the present paper, it is straightforward to argue (by considering instances in which each agent has at most two tiers of preference, where the first tier represents edges, and the second tier represents non-edges) that if a strictly Pareto-efficient mechanism admits an $O(f(n))$-time implementation, then the complexity of computing a maximum cardinality matching in a bipartite graph is $O(f(n))$. Given the foregoing remarks, it is natural to investigate whether the Hopcroft-Karp algorithm can be adapted to our setting to improve the $O(n^3)$ time bound established in Appendix E to $O(n^{2.5})$.

Acknowledgments

The author is grateful to Himanshu Chauhan, Onur Domaniç, Pravesh Kothari, and Chris Tosh for their many helpful comments on earlier versions of this work.

References

[1] D. J. Abraham, K. Cechlárová, D. F. Manlove, and K. Mehlhorn. Pareto-optimality in housing allocation problems. In *Proceedings of the 15th International Symposium on Algorithms and Computation, LNCS 3341*, pages 3–15, 2004. Also appears in LNCS 3827, with publisher typesetting errors corrected.

[2] J. Alcalde-Unzu and E. Molis. Exchange of indivisible goods and indifferences: The Top Trading Absorbing Sets mechanisms. *Games and Economic Behavior*, 73:1–16, 2011.

[3] A. Aziz and B. de Keijzer. Housing markets with indifferences: A tale of two mechanisms. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, pages 1249–1255, July 2012.

[4] C. G. Bird. Group incentive compatibility in a market with indivisible goods. *Economic Letters*, 14:309–313, 1984.

[5] J. E. Hopcroft and R. M. Karp. An $n^{5/2}$ algorithm for maximum matching in bipartite graphs. *SIAM Journal on Computing*, 2:225–231, 1973.

[6] P. Jaramillo and V. Manjunath. The difference indifference makes in strategy-proof allocation of objects. *Journal of Economic Theory*, 147:1913–1946, 2012.

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

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

[9] A. E. Roth. Incentive compatibility in a market with indivisible goods. *Economic Letters*, 9:127–132, 1982.

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

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

[12] J. Wako. A note on the strong core of a market with indivisible goods. *Journal of Mathematical Economics*, 13:189–194, 1984.

A Trading and Distance

The purpose of this section is to establish Lemma A.1, which immediately implies Lemma 3.4.

Lemma A.1. *Let* $G = (U, V, E)$ *be a configuration, let* C *belong to* $cycles(G)$*, and let* G' *denote* $trade(G, C)$*. For any nonnegative integer* k*, let* $P(k)$ *denote the predicate "for any house* v *in* V *such that* $distance(G', v) = 2k + 1$*, the following two conditions hold: (1) if* v *does not belong to* C *then* $distance(G, v) \leq 2k + 1$*; (2) if* v *belongs to* C *then* $distance(G, v) \leq 2k - 1$*."*

Proof. We use induction to prove that $P(k)$ holds for all nonnegative integers k.

Base case ($k = 0$). Let v be a house in V such that $distance(G', v) = 1$. Lemma 3.3 implies that v does not belong to C. Let u denote $agent(G', v)$. Since $distance(G', v) = 1$, we know that u belongs to $unsatisfied(G')$. Since v does not belong to C, we have $agent(G, v) = u$. Since u belongs to $unsatisfied(G')$, $\Gamma(G', u) = \Gamma(G, u)$, and $house(G', u) = house(G, u)$, we deduce that u belongs to $unsatisfied(G)$. Hence $distance(G, v) = 1$, as required.

Induction step. Let k be a nonnegative integer, and assume that $P(k)$ holds. We need to prove that $P(k + 1)$ holds. Let v be a house in V such that $distance(G', v) = 2(k + 1) + 1 = 2k + 3$. Let u denote $agent(G', v)$. Since $distance(G', v) = 2k + 3$ and $agent(G', v) = u$, there is a house v' in $\Gamma(G', u)$ such that $distance(G', v') = 2k + 1$. The induction hypothesis implies that $distance(G, v') \leq 2k + 1$. We now consider two cases.

Case 1: v does not belong to C. Thus $agent(G, v) = agent(G', v) = u$. Since $\Gamma(G', u) = \Gamma(G, u)$, there is a path of length two from v to v' in G, and hence $distance(G, v) \leq 2k + 3$, as required.

Case 2: v belongs to C. Thus $next(G, u) = v$. Since $\Gamma(G', u) = \Gamma(G, u)$, we know that v' belongs to $\Gamma(G, u)$. Since $next(G, u) = v$ and v' belongs to $\Gamma(G, u)$, we deduce that $distance(G, v) \leq distance(G, v')$. Since $distance(G, v') \leq 2k + 1$, we have $distance(G, v) \leq 2k + 1$, as required. □

B Edge Revelation

The purpose of this section is to establish Lemma B.2, which we use to prove Lemma 4.1.

Based on Lemma 3.1, we know that if $distance(G, v)$ is finite, then the unique outgoing path of length $distance(G, v)$ starting at v in $pruned(G)$ is a shortest path in G from v to an agent in

$\mathit{unsatisfied}(G)$. We define this agent as $\mathit{attractor}(G, v)$. If $\mathit{distance}(G, v) = \infty$ then we define $\mathit{attractor}(G, v)$ as nil.

Lemma B.1. *Let $G = (U, V, E)$ be a configuration, let u belong to $\mathit{unsatisfied}(G)$, let v belong to V, and let G' denote $(U, V, E + (u, v))$. Then $G \lesssim G'$ and the following claims hold.*

1. *If u belongs to $\mathit{unsatisfied}(G')$, then for any house v' in V, we have $\mathit{attractor}(G', v') = \mathit{attractor}(G, v')$ and $\mathit{distance}(G', v') = \mathit{distance}(G, v')$.*

2. *If u belongs to $\mathit{satisfied}(G')$, then for any house v' in V such that $\mathit{attractor}(G, v') \neq u$, we have $\mathit{distance}(G', v') = \mathit{distance}(G, v')$.*

Proof. In order to establish that $G \lesssim G'$ holds, the only nontrivial conjunct to be shown is that $\mathit{distance}(G', v') \geq \mathit{distance}(G, v')$ for all houses v' in V.

Case 1: u belongs to $\mathit{unsatisfied}(G')$. Hence $u \neq \mathit{agent}(G, v)$ and the edge (u, v) does not belong to a shortest path in G' from any house v' to an agent in $\mathit{unsatisfied}(G')$. It follows that for any house v' in V, we have $\mathit{distance}(G', v') = \mathit{distance}(G, v')$. Hence $G \lesssim G'$ holds. For any agent u' in $U - u$, we have $\mathit{next}(G', u') = \mathit{next}(G, u')$ since $\Gamma(G', u') = \Gamma(G, u')$ and $\mathit{distance}(G', v') = \mathit{distance}(G, v')$ for all v' in V. Hence $\mathit{pruned}(G')$ is the same as $\mathit{pruned}(G)$ except that $\mathit{next}(G', u)$ may differ from $\mathit{next}(G, u)$. Since u belongs to $\mathit{unsatisfied}(G')$, we deduce that $\mathit{attractor}(G', v') = \mathit{attractor}(G, v')$ for all houses v' in V.

Case 2: u belongs to $\mathit{satisfied}(G')$. Hence $u = \mathit{agent}(G, v)$. Let v' be an arbitrary house in V. We need to prove that $\mathit{distance}(G', v') \geq \mathit{distance}(G, v')$ and that this inequality is tight if $\mathit{attractor}(G, v') \neq u$.

Case 2.1: $\mathit{distance}(G', v') = \infty$. Thus $\mathit{distance}(G', v') \geq \mathit{distance}(G, v')$. It remains to prove that if $\mathit{attractor}(G, v') \neq u$, then $\mathit{distance}(G, v') = \infty$.

Case 2.1.1: $\mathit{attractor}(G, v') = \mathit{nil}$. Thus $\mathit{distance}(G, v') = \infty$, as required.

Case 2.1.2: $\mathit{attractor}(G, v') = u'$ for some agent u' in $U - u$. Then there is a path P in G from v' to u' such that u does not appear on P. It follows that path P also exists in G'. Since u' belongs to the set $\mathit{unsatisfied}(G')$, which is contained in the set $\mathit{unsatisfied}(G)$, we conclude that $\mathit{distance}(G', v')$ is finite, contradicting the Case 2.1 assumption.

Case 2.2: $\mathit{distance}(G', v')$ is finite. Let P be a shortest path in G' from v' to an agent in $\mathit{unsatisfied}(G')$; thus P is of length $\mathit{distance}(G', v')$. We need to argue that $\mathit{distance}(G, v')$ is at most the length of P. If u does not belong to path P, then P is a path in G from v' to an agent in $\mathit{unsatisfied}(G)$, and so $\mathit{distance}(G, v')$ is at most the length of P. If u belongs to path P, then let P' denote the prefix of P terminating at u, and observe that P' is a path in G from v' to an agent in $\mathit{unsatisfied}(G)$; hence $\mathit{distance}(G, v')$ is at most the length of P', which is at most the length of P. Hence $G \lesssim G'$ holds.

It remains to argue that if $\mathit{attractor}(G, v') \neq u$, then $\mathit{distance}(G', v') = \mathit{distance}(G, v')$. We have already established that $\mathit{distance}(G, v')$ is finite; hence $\mathit{attractor}(G, v') \neq \mathit{nil}$. Let u^* denote $\mathit{attractor}(G, v')$, and assume that $u^* \neq u$. Let P denote the unique path of length $\mathit{distance}(G, v')$ in $\mathit{pruned}(G)$ from v' to u^*. Since u^* belongs to $\mathit{unsatisfied}(G)$ and $u^* \neq u$, we conclude that u^* belongs to $\mathit{unsatisfied}(G')$. Since P is a path in $\mathit{pruned}(G)$, which is a subgraph of G', we conclude that P exists in G'. Since P exists in G', we deduce that $\mathit{distance}(G', v') \leq \mathit{distance}(G, v')$. Since we have already established that $\mathit{distance}(G', v') \geq \mathit{distance}(G, v')$, we conclude that $\mathit{distance}(G', v') = \mathit{distance}(G, v')$, as required. ☐

Lemma B.2. *Let $G = (U, V, E)$ be configuration, let u be an agent in $\mathit{unsatisfied}(G)$, let E' be a set of edges in $\{u\} \times V$, and let G' denote the configuration $(U, V, E \cup E')$. Then $G \lesssim G'$ and the following claims hold.*

1. *If u belongs to $\mathit{unsatisfied}(G')$, then for any house v in V, we have $\mathit{distance}(G', v) = \mathit{distance}(G, v)$.*

2. *If u belongs to $\mathit{satisfied}(G')$, then for any house v in V such that $\mathit{attractor}(G, v) \neq u$, we have $\mathit{distance}(G', v) = \mathit{distance}(G, v)$.*

Proof. Let E'' denote $E' - (u, \mathit{house}(G, u))$, and let G'' denote the configuration $(U, V, E \cup E'')$. Then $G \lesssim G''$ holds by repeated application of Lemmas B.1 and 3.8. Furthermore, u belongs to $\mathit{unsatisfied}(G'')$, and by repeated application of Lemma B.1, we have $\mathit{attractor}(G'', v) = \mathit{attractor}(G, v)$ and $\mathit{distance}(G'', v) = \mathit{distance}(G, v)$ for all houses v in V. If $E'' = E'$, this completes the proof.

It remains to consider the case where $E'' \neq E'$. In this case, u belongs to $\mathit{satisfied}(G')$, and by an additional application of Lemma B.1, we find that $G'' \lesssim G'$ holds and $\mathit{distance}(G', v) = \mathit{distance}(G'', v)$ for all houses v in V such that $\mathit{attractor}(G'', v) \neq u$. Since we have established above that $\mathit{attractor}(G'', v) = \mathit{attractor}(G, v)$ for all houses v in V, we conclude that $\mathit{distance}(G', v) = \mathit{distance}(G'', v)$ for all houses v in V such that $\mathit{attractor}(G, v) \neq u$.

Since $G \lesssim G''$ and $G'' \lesssim G'$, Lemma 3.8 implies $G \lesssim G'$, as required. Let v be a house in V such that $\mathit{attractor}(G, v) \neq u$. In the foregoing we have established that $\mathit{distance}(G'', v) = \mathit{distance}(G, v)$ and $\mathit{distance}(G', v) = \mathit{distance}(G'', v)$. Thus $\mathit{distance}(G', v) = \mathit{distance}(G, v)$, as required. □

C Confluence

The six lemmas below are used to prove Lemma 4.4.

Lemma C.1. *Let $G = (U, V, E)$ be a configuration, let C and C' be distinct cycles in $\mathit{cycles}(G)$, and let G' denote $\mathit{trade}(G, C)$. Then C' belongs to $\mathit{cycles}(G')$.*

Proof. Since each house v in V has outdegree one in $\mathit{pruned}(G)$, the cycles C and C' are disjoint. Let u be an arbitrary agent on C'. Thus $\mathit{next}(G, u) \neq \mathit{nil}$. Let v denote the house $\mathit{next}(G, u)$, which is on C'. Lemma 3.2 implies that there is at least one agent in $\mathit{unsatisfied}(G)$ on C', and hence that $\mathit{distance}(G, v)$ is finite. Grow a path P in C' by starting at v and following edges of C' until an agent in $\mathit{unsatisfied}(G)$ is reached. Lemma 3.1 implies that P is of length $\mathit{distance}(G, v)$. Since P is a portion of the cycle C', we deduce that P is disjoint from C, and hence that P exists in G'. It follows that $\mathit{distance}(G', v) \leq \mathit{distance}(G, v)$. On the other hand, Lemma 3.4 implies that $\mathit{distance}(G', v) \geq \mathit{distance}(G, v)$. We conclude that $\mathit{distance}(G', v) = \mathit{distance}(G, v)$. By Lemma 3.10, we have $G \lesssim G'$. Since $G \lesssim G'$, $\mathit{distance}(G', v) = \mathit{distance}(G, v)$, and $\Gamma(G', u) = \Gamma(G, u)$, Lemma 3.9 implies that $\mathit{next}(G', u) = \mathit{next}(G, u)$.

Since $\mathit{next}(G', u) = \mathit{next}(G, u)$ for all agents u on C', and $\mathit{agent}(G', v) = \mathit{agent}(G, v)$ for all houses v on C', the claim of the lemma follows. □

Lemma C.2. *Let W be a wpp, let G belong to $\mathit{configs}(W)$, let C belong to $\mathit{cycles}(G)$, and let G' belong to $\Gamma(W, G) - \mathit{trade}(G, C)$. Then C belongs to $\mathit{cycles}(G')$.*

Proof. There are two cases to consider.

Case 1: The configuration G' is of the form $\mathit{trade}(G, C')$ for some C' in $\mathit{cycles}(G) - C$. In this case, the claim of the lemma follows from Lemma C.1.

Case 2: The configuration G' is of the form $\mathit{reveal}(W, G, u_0)$ for some agent u_0 in U. Thus u_0 belongs to $\mathit{exhausted}(G)$, and hence u_0 is not on C. Thus Lemma 4.1 implies that $\mathit{next}(G', u) = \mathit{next}(G, u)$ holds for all agents u on C. Furthermore, $\mathit{agent}(G', v) = \mathit{agent}(G, v)$ for all houses v in V. The claim of the lemma follows. □

Lemma C.3. *Let W be a wpp, let G belong to configs(W), let u belong to $\mathit{exhausted}(G)$, and let G' belong to $\Gamma(W, G) - \mathit{reveal}(W, G, u)$. Then u belongs to $\mathit{exhausted}(G')$.*

Proof. We claim that $\Gamma(G', u) = \Gamma(G, u)$. If the configuration G' is of the form $\mathit{trade}(G, C)$ for some C in $\mathit{cycles}(G)$, then $\Gamma(G', u') = \Gamma(G, u')$ for all agents u' in U, and the claim holds. Otherwise, the configuration G' is of the form $\mathit{reveal}(W, G, u_0)$ for some agent u_0 in $U - u$, and since $u \neq u_0$, the claim holds.

Since u belongs to $\mathit{exhausted}(G)$, we have $\mathit{distance}(G, v) = \infty$ for all houses v in $\Gamma(G, u)$. Lemma 4.2 implies $G \lesssim G'$, and hence that $\mathit{distance}(G', v) \geq \mathit{distance}(G, v)$ for all houses v in V. Since $\Gamma(G', u) = \Gamma(G, u)$, we conclude that $\mathit{distance}(G', v) = \infty$ for all houses v in $\Gamma(G', u)$, and hence that $\mathit{next}(G', u) = \mathit{nil}$.

We claim that $\mathit{house}(G', u) = \mathit{house}(G, u)$. If the configuration G' is of the form $\mathit{trade}(G, C)$ for some C in $\mathit{cycles}(G)$, then u is not on C since $\mathit{next}(G, u) = \mathit{nil}$; hence the claim holds. Otherwise, the configuration G' is of the form $\mathit{reveal}(W, G, u_0)$ for some agent u_0 in $U - u$; hence $\mathit{allocation}(G') = \mathit{allocation}(G)$ and the claim holds.

Since $\mathit{house}(G', u) = \mathit{house}(G, u)$, $\Gamma(G', u) = \Gamma(G, u)$, and u belongs to $\mathit{unsatisfied}(G)$, we conclude that u belongs to $\mathit{unsatisfied}(G')$. Since u belongs to $\mathit{unsatisfied}(G')$ and $\mathit{next}(G', u) = \mathit{nil}$, we conclude that u belongs to $\mathit{exhausted}(G')$, as required. □

Lemmas C.1, C.2 and C.3 ensure that all of the expressions appearing in the next three lemma statements are well-defined.

Lemma C.4. *Let $G = (U, V, E)$ be a configuration, and let C and C' be distinct cycles in $\mathit{cycles}(G)$. Then $\mathit{trade}(\mathit{trade}(G, C), C') = \mathit{trade}(\mathit{trade}(G, C'), C)$.*

Proof. Since each agent or house in $\mathit{pruned}(G)$ has outdegree 1, the cycles C and C' are disjoint. The claim of the lemma follows easily. □

Lemma C.5. *Let W be a wpp, let G belong to configs(W), let u belong to $\mathit{exhausted}(G)$, and let C belong to $\mathit{cycles}(G)$. Then $\mathit{trade}(\mathit{reveal}(W, G, u), C) = \mathit{reveal}(W, \mathit{trade}(G, C), u)$.*

Proof. Straightforward. □

Lemma C.6. *Let W be a wpp, let G belong to configs(W), and let u and u' be distinct agents in $\mathit{exhausted}(G)$. Then $\mathit{reveal}(W, \mathit{reveal}(W, G, u), u') = \mathit{reveal}(W, \mathit{reveal}(W, G, u'), u)$.*

Proof. Straightforward. □

D Admissibility

The two lemmas below are used to prove Lemma 4.6.

Lemma D.1. *Let* $W = (U, V, \succsim)$ *be a wpp, let* G *belong to* $admissible(W)$*, and let* G' *belong to* $\Gamma(W, G)$*. Then* G' *belongs to* $admissible(W)$*.*

Proof. Let u belong to U, let v denote $house(G, u)$, and let v' denote $house(G', u)$. Lemma 4.2 implies $G \lesssim G'$.

Case 1: u belongs to $frozen(G)$. Since G belongs to $admissible(W)$, we find that $bottom(W, G', u, v)$ holds. Since $G \lesssim G'$, we find that u belongs to $frozen(G')$, $\Gamma(G', u) = \Gamma(G, u)$, and $v' = v$. Hence $bottom(W, G', u, v')$ holds.

Case 2: u does not belong to $frozen(G)$. Let V_0 denote the set of all houses v_0 in $\Gamma(G, u)$ such that $agent(G, v_0)$ does not belong to $frozen(G)$. Since G belongs to $admissible(W)$, we have $bottom(W, G, u, v_0)$ for all houses v_0 in V_0. We consider two subcases.

Case 2.1: u does not belong to $frozen(G')$. We need to prove that for any house v in $\Gamma(G', u)$ such that $agent(G', v)$ does not belong to $frozen(G')$, we have $bottom(W, G', u, v)$. Since $G \lesssim G'$, we find that $\Gamma(G', u)$ contains $\Gamma(G, u)$ and $frozen(G')$ contains $frozen(G)$. If $\Gamma(G', u) = \Gamma(G, u)$, the desired claim follows immediately. Otherwise, $G' = reveal(W, G, u)$ and hence u belongs to $exhausted(G)$. Thus $agent(G, v_1)$ belongs to $frozen(G)$ (and hence also $frozen(G')$) for all houses v_1 in $\Gamma(G, u)$, and the desired claim follows easily from the definition of $reveal(W, G, u)$.

Case 2.2: u belongs to $frozen(G')$. Thus u belongs to $satisfied(G')$. We need to establish $bottom(W, G', u, v')$. We consider two subcases.

Case 2.2.1: V_0 is nonempty. Thus $\Gamma(G', u) = \Gamma(G, u)$. Since u belongs to $satisfied(G')$ and $G \lesssim G'$, we deduce that v' belongs to V_0. Since v' belongs to V_0, we have $bottom(W, G, u, v')$. We conclude that $bottom(W, G', u, v')$ holds, as required.

Case 2.2.2: V_0 is empty. Thus u belongs to $exhausted(G)$. Since u belongs to $satisfied(G')$, we conclude that $G' = reveal(W, G, u)$ and $v' = v$. The desired claim follows easily from the definition of $reveal(W, G, u)$. □

Lemma D.2. *Let* $W = (U, V, \succsim)$ *be a wpp, let* G *belong to* $admissible(W)$*, let* G' *belong to* $\Gamma(W, G)$*, let* u *belong to* U*, let* v *denote* $house(G, u)$*, and let* v' *denote* $house(G', u)$*. Then* $v' \succsim_u v$*. Furthermore, if* u *belongs to* $satisfied(G)$ *then* $v' \sim_u v$*.*

Proof. If $v = v'$ then the claim of the lemma is trivial. For the remainder of the proof, assume that $v \neq v'$. Thus $G' = trade(G, C)$ for some C in $cycles(G)$ such that agent u is on C. Let u' denote $agent(G, v')$. Since no agents in $frozen(G)$ appear on C, neither u nor u' belongs to $frozen(G)$. Since v' is equal to $next(G, u)$, we conclude that v' belongs to $\Gamma(G, u)$. Since v' belongs to $\Gamma(G, u)$, u' does not belong to $frozen(G)$, and G belongs to $admissible(W)$, we conclude that $bottom(W, G, u, v')$ holds. If u belongs to $unsatisfied(G)$, then since G belongs to $configs(W)$ and $bottom(W, G, u, v')$ holds, we have $v' \succ_u v$. If u belongs to $satisfied(G)$, then v belongs to $\Gamma(G, u)$. Since v belongs to $\Gamma(G, u)$, u does not belong to $frozen(G)$, and G belongs to $admissible(W)$, we conclude that $bottom(W, G, u, v)$ holds, and hence that $v \sim_u v'$. □

E A Fast Implementation

In this section we describe an $O(n^3)$-time deterministic algorithm, Algorithm 2. It will be evident that any execution of Algorithm 2 corresponds to a possible execution of Algorithm 1. Thus Theorem 1 implies that Algorithm 2 has the same input-output behavior as Algorithm 1.

Let $W = (U, V, \succsim)$ be a wpp and let G be a non-final configuration in $admissible(W)$. Below we describe an $O(|V|^2)$ time subroutine to compute a configuration G' in $\Gamma^*(W, G)$ such that $|unsatisfied(G')| < |unsatisfied(G)|$. By Lemma 4.6, we find that G' belongs to $admissible(W)$. Thus we can iteratively apply this subroutine at most $|V|$ times, yielding an overall time bound of $O(|V|^3)$.

The $O(|V|^2)$-time subroutine works in three phases, as follows. In the first phase, we use a breadth-first traversal in the reversal of G (i.e., the graph obtained by reversing the direction of the edges in G), starting from the set of agents in $unsatisfied(G)$ (which are easy to identify), to compute $distance(G, v)$ for all houses v in V, and $next(G, u)$ for all agents u in U. Then we identify the set of agents in $exhausted(G)$, call it U', and the set of all houses v such that $distance(G, v)$ is finite, call it V'. The time complexity of the first phase is easily seen to be $O(|V|^2)$.

In the second phase, we process the agents in U' in arbitrary order. To process such an agent u, we repeatedly update G to $reveal(W, G, u)$ until $\Gamma(G, u) \cap V'$ is nonempty. Each such update merely involves adding some new outgoing edges to agent u. The total number of new edges added to u is at most $|V|$, and the time complexity of processing u is $O(|V|)$. Once u has been processed, we check whether u now belongs to $satisfied(G)$. If so, we can terminate the subroutine and take the desired configuration G' to be the current configuration G. (As an optimization, we can go ahead and process any remaining agents in U' before terminating the subroutine.) If we process all of the agents in U' without arriving at a configuration G' such that $|unsatisfied(G')| < |unsatisfied(G)|$, then we proceed to the third phase. Since at most $|V|$ agents are processed in the second phase, the time complexity of the second phase is $O(|V|^2)$.

In the third phase, since $exhausted(G)$ is empty and G is not final (since all of the agents in U' continue to belong to $unsatisfied(G)$), Lemma 3.6 implies that $cycles(G)$ is nonempty. As in a standard TTC algorithm for the simple case of strict preferences, we can easily identify all of the cycles in $cycles(G)$ in $O(|V|)$ time. We can then update G to $trade(G, C)$ for some C in $cycles(G)$. As argued in the proof of Lemma 4.3, such an update yields a configuration G' such that $|unsatisfied(G')| < |unsatisfied(G)|$. The time complexity of the third phase is $O(|V|)$. (As an optimization, we can process all of the cycles in $cycles(G)$; the time complexity of the third phase remains $O(|V|)$.)