Designing Truthful Mechanisms: A study of Voting and Matching Rules

A DISSERTATION PRESENTED
BY
ABHIGYAN BOSE
TO
THE ECONOMIC RESEARCH UNIT

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN THE SUBJECT OF
QUANTITATIVE ECONOMICS

INDIAN STATISTICAL INSTITUTE
KOLKATA, WEST BENGAL, INDIA
JULY 2025

Contents

Author List

The following authors contributed to Chapter 1: Abhigyan Bose, and Souvik Roy.

The following authors contributed to Chapter 2: Abhigyan Bose, and Souvik Roy.

The following authors contributed to Chapter 3: Abhigyan Bose, and Souvik Roy.

The following authors contributed to Chapter 4: Abhigyan Bose, and Souvik Roy.

The following authors contributed to Chapter 5: Abhigyan Bose, and Souvik Roy.

The following authors contributed to Chapter 6: Abhigyan Bose, and Souvik Roy.

Listing of figures

To all my teachers.

Acknowledgments

I would like to express my deepest and most respectful gratitude to my supervisor, Professor Souvik Roy whose unwavering support, insightful guidance, and steadfast mentorship have been instrumental in shaping both my research and my academic journey. I am especially grateful for his immense patience with my shortcomings and for his steady encouragement during moments of self-doubt and struggle. His intellectual rigor, uncompromising dedication to research, and the exceptional quality of his work have been a constant source of inspiration—qualities I deeply admire and hope to emulate in my own career. His meticulous attention to detail and genuine care for my growth as both a researcher and an individual have left a lasting impact on me.

I am particularly indebted to Professor Arunava Sen and Professor Debasis Mishra of the Indian Statistical Institute, Delhi, for their invaluable advice at various stages of my work. Their willingness to engage in lively discussions, suggest new directions, and share resources not only deepened my understanding of sophisticated models but also encouraged me to tackle unfamiliar problems with confidence. I wish to remember and pay tribute to Professor Manipushpak Mitra—whose recent demise is a profound loss to our community; his brilliance, kindness, and enduring contributions continue to inspire us all. I am grateful to the entire faculty of our department for their generous and continuous support in my journey.

This journey would have been far less vibrant without my colleagues at the Economic Research Unit: Sayan, Chirayata, Aarvind, and Arun,—whose conversations, often veering well beyond technical matters, made our workplace both stimulating and warm. Special thanks to Ujjwal da, Soumyarup da, and Maduparna di, who acted as quasi co-guides—helping me navigate the hurdles of the Ph.D. path with their seasoned experience. I also appreciate every member of the ERU department—office staff and junior and senior research fellows alike, whose everyday camaraderie and practical support made each challenge feel surmountable.

I am grateful to my dear friends Arkav, Debojyoti, and Arijit for their patience and for always lending an ear to my endless musings on the trials and triumphs of Ph.D. life. Your companionship has been a source of comfort and sanity throughout this demanding journey.

Finally, I remain forever indebted to my parents, dada, and boudi, whose unconditional love, and unwavering belief in me have been the cornerstone of every success. Their constant care and support have helped me stay grounded and resilient. And to 'Kuttush,' my beloved cat, who arrived during my Ph.D. years—thank you for the endless joy, laughter, and warmth your playful antics have brought into my life.

1
Introduction

This thesis consists of six chapters related to voting and matching theory. The first three chapters are dedicated to voting theory, while the remaining three chapters focus on matching theory. A brief introduction of the chapters are provided below.

1.1 "Incentive-compatible voting rules with positively correlated beliefs": Correction

In this chapter, we provide a correction of an result in the existing literature. Theorem 1 in [5] provides a necessary condition for a social choice function to be LOBIC with respect to a belief system satisfying top-set (TS) correlation. In this chapter, we provide a counter example to that theorem and consequently provide a new necessary condition for the same in terms of sequential ordinal nondomination.

1.2 Ordinal Bayesian incentive-compatible voting rules with correlated belief under betweenness property

In this chapter, we consider social choice functions (SCFs) that are locally robust ordinal Bayesian incentive compatible (LOBIC) with respect to correlated priors. We model such priors using a betweenness property and assume the coexistence of both positively and negatively correlated priors. We introduce the notion of strong ordinal non domination (strong OND) and show that strong OND is a sufficient condition for an SCF to be LOBIC in this framework.

1.3 Social choice with network-based common knowledge

In this chapter, we study mechanism design in environments where agents possess *partial common knowledge* of preferences, structured through an underlying network. In our model, each agent observes the preferences of their neighbors, and this informational structure is commonly known to all agents and the mechanism designer. Each agent reports both their own preferences and their knowledge of their neighbors' preferences. A mechanism then selects an outcome based on the entire profile of such reports. We investigate when a standard social choice function can be implemented via an *incentive-compatible* mechanism under various assumptions on the network structure. First, we show that when each agent is connected to at least two others, every social choice function is extendable to an incentive-compatible mechanism. Second, in the minimal case of two agents connected by a single link, we provide a complete characterization of extendability in terms of a novel condition called the *Lower Contour Set Intersection Property* (LCSIP). Finally, for arbitrary connected networks, we introduce a generalization of LCSIP and prove that it is both necessary and sufficient for incentive-compatible extension.

1.4 Matching under multiple single-peaked domains

In this chapter, we study the classical housing allocation problem in the Shapley-Scarf model under a broad class of restricted preference domains known as *multiple single-peaked preferences*. The canonical *Crawler* algorithm, designed for single-peaked preferences, is not well-defined in this setting due to the absence of a common ordering over certain subsets of houses. To address this, we introduce the *C-TTC algorithm*, a novel matching rule that blends the structure of the Crawler with the flexibility of the Top Trading Cycles (TTC) algorithm. We show that C-TTC satisfies three fundamental properties—*individual rationality* (IR), *Pareto optimality* (PO), and *strategy-proofness* (SP)—across the entire class of multiple single-peaked domains. Our mechanism generalizes both TTC (when preferences are unrestricted) and Crawler (when preferences are single-peaked), providing a unified framework for allocation in partially structured environments. Furthermore, we investigate the stronger incentive requirement of *obvious strategy-proofness* (OSP). While OSP is achievable in single-peaked domains, we prove that no matching rule satisfies IR, PO, and OSP in any non-trivial multiple single-peaked domain.

1.5 Matching under tree single-peaked preferences

This chapter investigates the design of matching mechanisms in the classical Shapley-Scarf housing market. We focus on *tree single-peaked preferences*, a natural generalization of single-peaked preferences on a line, and introduce the *Tree Neighborhood Algorithm* (TN algorithm) tailored to this domain. We show that the TN algorithm satisfies *individual rationality* (IR), *Pareto optimality* (PO), and *strategy-proofness* for all tree single-peaked domains. Furthermore, we prove that no IR and PO matching rule satisfies *obvious strategy-proofness* (OSP) in this setting unless the underlying tree is a line. When the tree is a line graph, the TN algorithm generalizes the well-known Crawler algorithm and retains OSP.

1.6 Matching rules and obvious strategy-proofness

This chapter provides a rigorous characterization of matching rules in one-to-one, one-sided matching markets that simultaneously satisfy individual rationality (IR), Pareto optimality (PO), and obvious strategy-proofness (OSP). We introduce a novel necessary condition on preference domains, termed *bottom freeness*, which is fundamental for the existence of mechanisms meeting these criteria. Utilizing this condition, we establish several impossibility results, proving the non-existence of mechanisms that are IR, PO, and OSP on domains such as the circular domain, the union of single-peaked and single-dipped domains, and the unrestricted domain. Moreover, we examine the Top Trading Cycles

(TTC) mechanism, a prominent rule in matching theory, and demonstrate that it fails to satisfy OSP even on regular single-peaked domains, thereby extending previous negative findings in [2]. To circumvent this limitation, we introduce a novel domain restriction, termed the *almost top property*, and prove that under this condition, the TTC mechanism is OSP-implementable.

2

“Incentive-compatible voting rules with positively correlated beliefs”: Correction

2.1 Introduction

A social choice function (SCF) selects an alternative at every collection of preferences of the agents in a society. An SCF is called ordinal Bayesian incentive compatible (OBIC) with respect to a belief if, by misreporting his sincere preference, no agent can increase his expected utility according to his belief for any utility function representing his sincere preference. An SCF is called locally robust OBIC (LOBIC) with respect to a belief if it is OBIC with respect to all beliefs lying in a small neighborhood of the original belief. LOBIC ensures that agents are incentivized to reveal their sincere preferences even if the designer is *slightly* unsure about their beliefs.

Theorem 1 in [5] says that a unanimous SCF is LOBIC with respect to a TS-(top-set) correlated belief if and only if it satisfies a property called ordinal nondomination (OND). We show that the "only if" part of this theorem is not correct, and consequently we provide a necessary condition for LOBIC in terms of sequential ordinal nondomination (sequential OND).[1]

It is worth emphasizing that we do not put any restriction on the beliefs of the agents.[2] In particular, beliefs are not required to be independent or even common.[3]

2.2 A Counter Example to Theorem 1 in [5]

Theorem 1 in [5] says that every TS-LOBIC SCF satisfies OND. Later, they remark that the statement holds for every LOBIC SCF. In what follows, we provide a counter example to this statement. We consider beliefs satisfying top-set (TS) correlation to clarify the fact that the result does not hold for TS-LOBIC SCFs also. Before proceeding to the counter example, let us recall the following definitions from [5].

Let A be a finite set of alternatives and let $N = \{1, \ldots, n\}$ be a set of n agents. We denote by $\mathbb{P}$ the set of all (strict) preferences on A. A social choice function (SCF) is a mapping $f : \mathbb{P}^n \to A$. A belief μ_i for agent i is a probability distribution on the set $\mathbb{P}^n$, i.e., it is a map $\mu_i : \mathbb{P}^n \to [0, 1]$ such that $\sum_{P \in \mathbb{P}^n} \mu_i(P) = 1$. The utility function $u : A \to \mathbb{R}$ represents $P_i \in \mathbb{P}$ if and only if for all $a, b \in A$, we have $aP_ib \iff u(a) > u(b)$.

Definition 2.2.1 *A belief for an agent i, μ_i is positively TS-correlated if for all $P_i \in \mathbb{P}$, all $k = 1, \ldots m-1$, and all $D \subset A$ such that $D \neq B_k(P_i)$* [4] *and $|D| = k$, we have*

$$\sum_{P_{-i} | B_k(P_j) = B_k(P_i) \;\forall\, j \neq i} \mu_i(P_{-i}|P_i) > \sum_{P_{-i} | B_k(P_j) = D \;\forall\, j \neq i} \mu_i(P_{-i}|P_i). \tag{2.1}$$

Definition 2.2.2 *An SCF $f : \mathbb{P}^n \to A$ is OBIC with respect to belief system $\mu_N = (\mu_1, \ldots, \mu_n)$ if for all $i \in N$, for all integers $k = 1, \ldots, m$, and for all $P_i, P_i' \in \mathbb{P}$, we have*

$$\sum_{P_{-i} | f(P_i, P_{-i}) \in B_k(P_i)} \mu_i(P_{-i} \mid P_i) \geq \sum_{P_{-i} | f(P_i', P_{-i}) \in B_k(P_i)} \mu_i(P_{-i} \mid P_i). \tag{2.2}$$

[1] In the proof of Theorem 1 in [5], the authors consider two cases. However, to our understanding, there is a third case that the authors have missed.

[2] Although the statement of Theorem 1 in [5] involves beliefs that satisfy top-set correlation, they have remarked that the "only if" part of the theorem is more general as it holds for arbitrary beliefs. Since we deal with the "only if" part of this theorem, we present our result for arbitrary beliefs.

[3] See [16] and [17] for details on LOBIC SCFs under independent priors.

[4] Recall that $B_k(P_i)$ denotes top k alternatives in the ordering P_i.

Definition 2.2.3 *An SCF $f : \mathbb{P}^n \to A$ is locally robust OBIC (LOBIC) with respect to the belief system μ_N if there exists $\varepsilon > 0$ such that f is OBIC with respect to all μ'_N such that $\mu'_N \in B_\varepsilon(\mu_N)$.* [5]

Definition 2.2.4 *An SCF $f : \mathbb{P}^n \to A$ is TS-locally robust OBIC (TS-LOBIC) with respect to a belief system μ_N if*

(i) μ_i satisfy TS correlation for all $i \in N$.

(ii) f is LOBIC with respect to μ_N.

Definition 2.2.5 *An SCF $f : \mathbb{P}^n \to A$ satisfies ordinal nondomination (OND) if for all $i \in N$, for all $P_i, P'_i \in \mathbb{P}$, and all $P_{-i} \in \mathbb{P}^{n-1}$ such that $f(P'_i, P_{-i}) P_i f(P_i, P_{-i})$, there exists $P'_{-i} \in \mathbb{P}^{n-1}$ such that the following statements hold:*

(i) Either $f(P_i, P'_{-i}) = f(P'_i, P_{-i})$ or $f(P_i, P'_{-i}) P_i f(P'_i, P_{-i})$.

(ii) Either $f(P_i, P_{-i}) = f(P'_i, P'_{-i})$ or $f(P_i, P_{-i}) P_i f(P'_i, P'_{-i})$.

Example 2.2.6 *Suppose that there are two agents $\{1, 2\}$ and three alternatives $\{a, b, c\}$. Consider the domain $\mathbb{P}$ containing the set of all preferences over $\{a, b, c\}$. We denote by abc the preference where a, b, and c are the top-ranked, second-ranked, and third-ranked alternatives, respectively. In Table 2.2.1, we present an SCF, say $\hat{f}$, and in Table 2.2.2 and Table 2.2.3 we present the conditional beliefs μ_1 and μ_2 of agent 1 and agent 2, respectively. These tables are self-explanatory.*

1 \ 2	abc	acb	bac	bca	cab	cba
abc	a	a	c	a	b	b
acb	a	a	a	b	c	a
bac	b	a	b	b	a	c
bca	c	b	b	b	a	c
cab	a	a	b	c	c	c
cba	a	c	b	a	c	c

Table 2.2.1: Example of an SCF that does not satisfy OND but satisfies LOBIC

1 \ 2	abc	acb	bac	bca	cab	cba
abc	0.51	0.02	0.04	0.17	0.20	0.06
acb	0.02	0.51	0.20	0.01	0.09	0.17
bac	0.17	0.15	0.51	0.02	0.14	0.01
bca	0.15	0.17	0.02	0.51	0.01	0.14
cab	0.15	0.14	0.01	0.17	0.51	0.02
cba	0.01	0.21	0.23	0.01	0.03	0.51

Table 2.2.2: Conditional belief of Agent 1

1 \ 2	abc	acb	bac	bca	cab	cba
abc	0.51	0.02	0.01	0.01	0.01	0.09
acb	0.02	0.51	0.09	0.25	0.25	0.01
bac	0.09	0.17	0.51	0.02	0.15	0.17
bca	0.01	0.01	0.02	0.51	0.06	0.20
cab	0.17	0.20	0.17	0.20	0.51	0.02
cba	0.20	0.09	0.20	0.01	0.02	0.51

Table 2.2.3: Conditional belief of Agent 2

We claim the following facts about the SCF $\hat{f}$ and the prior beliefs μ_1 and μ_2. Claim 2.2.1 and Claim 2.2.2 establish that the SCF $\hat{f}$ is TS-LOBIC with respect to (μ_1, μ_2), while Claim 2.2.3 says that $\hat{f}$ does not satisfy the OND property. This contradicts Theorem 1 in [5].

Claim 2.2.1 *The conditional beliefs μ_1 and μ_2 satisfy top-set correlation.*

Proof: Recall that a belief system μ_N is TS-(top-set) correlated if for all $i \in N$, all $P_i \in \mathbb{P}$, all $k = 1, \dots m-1$, and all $D \subset A$ such that $D \neq B_k(P_i)$ and $|D| = k$, we have

$$\sum_{P_{-i}|B_k(P_j)=B_k(P_i) \ \forall\ j\neq i} \mu_i(P_{-i}|P_i) > \sum_{P_{-i}|B_k(P_j)=D \ \forall\ j\neq i} \mu_i(P_{-i}|P_i). \tag{2.3}$$

Observe from Tables 2.2.2 and 2.2.3 that for all $i = \{1, 2\}$ and all $P_i \in \mathbb{P}$, $P_{-i} = P_i$ implies $\mu_i(P_{-i}|P_i) = 0.51$. Since preferences P_{-i} such that $P_{-i} = P_i$ will always appear only in the left hand side of Inequality (2.3) and the corresponding probability is more than 0.5, it follows that Inequality (2.3) will always be satisfied by the belief system (μ_1, μ_2). This shows that (μ_1, μ_2) satisfy top-set correlation.

■

Claim 2.2.2 *The SCF $\hat{f}$ is TS-LOBIC with respect to $\mu_N = (\mu_1, \mu_2)$.*

The proof of this Claim is relegated to Appendix 2.4.

Claim 2.2.3 *The SCF $\hat{f}$ does not satisfy the OND property.*

Proof: Consider $P_1 = abc$, $P'_1 = acb$, and $P_2 = bac$. We have $\hat{f}(P'_1, P_2) P_1 \hat{f}(P_1, P_2)$. However, there is no P'_2 such that $\hat{f}(P_1, P_2) P_1 \hat{f}(P'_1, P'_2)$ or $\hat{f}(P_1, P_2) = \hat{f}(P'_1, P'_2)$ and $\hat{f}(P_1, P'_2) P_1 \hat{f}(P'_1, P_2)$ or $\hat{f}(P_1, P'_2) = \hat{f}(P'_1, P_2)$. Therefore, $\hat{f}$ does not satisfy the OND property. ■

This completes the verification that Example 2.2.6 is indeed a counter example to Theorem 1 in [5].

[5] The function $B_\varepsilon(\mu_i)$ denotes the open ball of radius ε centered at μ_i.

2.3 A necessary condition for LOBIC with respect to a(ny) correlated belief

In this section, we provide a necessary condition for an SCF to be LOBIC with respect to an arbitrary correlated belief. Our necessary condition uses the notion of sequential ordinal non-domination (sequential OND). In contrast to the OND property where a gain of agent i by manipulation can be paid back at *exactly one* preference profile P'_{-i}, in case of sequential OND the same can happen through a sequence of preference profiles $(P^1_{-i}, \dots, P^k_{-i})$ for some $k \geq 1$. Note that OND is a special case of sequential OND where the length of the sequence is 1. We use the following notation to facilitate the formal definition of sequential OND: for a preference P we use the notation R to denote the weak part of it, that is, aRb implies either aPb or $a = b$.

Definition 2.3.1 *For an SCF $f : \mathbb{P}^n \to A$ and a pair of distinct preferences (P_i, P'_i) in $\mathbb{P}$, a sequence $(P^1_{-i}, \dots, P^k_{-i})$ of elements of $\mathbb{P}^{n-1}$ is called an **OND sequence** for f with respect to (P_i, P'_i) if for all $l = 1, \dots, k-1$, we have $f(P'_i, P^l_{-i}) P_i f(P'_i, P^{l+1}_{-i})$ and $f(P_i, P^{l+1}_{-i}) R_i f(P'_i, P^l_{-i})$.*

Definition 2.3.2 *An SCF $f : \mathbb{P}^n \to A$ satisfies the **sequential OND** property if for all $i \in N$, all $P_i, P'_i \in \mathbb{P}$, and all $P_{-i} \in \mathbb{P}^{n-1}$ with $f(P'_i, P_{-i}) P_i f(P_i, P_{-i})$, there exists an OND sequence $(P^1_{-i}, \dots, P^k_{-i})$ for f with respect to (P_i, P'_i) such that $f(P_i, P_{-i}) R_i f(P'_i, P^k_{-i})$, $f(P_i, P^1_{-i}) R_i f(P'_i, P_{-i})$, and $f(P'_i, P_{-i}) P_i f(P'_i, P^1_{-i})$.*

In what follows, we argue that the SCF $\hat{f}$ in Table 2.2.1 satisfies the sequential OND property.

Claim 2.3.1 *The SCF $\hat{f}$ satisfies the OND property (and hence, the sequential OND property) for all situations except the one where $P_1 = abc$, $P'_1 = acb$, and $P_2 = bac$.*

The proof of this Claim is relegated to Appendix 2.5.

For the case where $P_1 = abc$, $P'_1 = acb$, and $P_2 = bac$, consider the sequence $(P^1_2 = bca, P^2_2 = cab)$ of preferences of agent 2. Note that (i) $\hat{f}(P_1, P^1_2) R_1 \hat{f}(P'_1, P_2)$, (ii) $\hat{f}(P_1, P_2) R_1 \hat{f}(P'_1, P^2_2)$, (iii) $\hat{f}(P_1, P^2_2) R_1 \hat{f}(P'_1, P^1_2)$ and (iv) $\hat{f}(P'_1, P_2) P_1 \hat{f}(P'_1, P^1_2) P_1 \hat{f}(P'_1, P^2_2)$. Thus, $\hat{f}$ satisfies the sequential OND property.

Theorem 2.3.3 *An SCF is LOBIC with respect to some belief system only if it satisfies the sequential OND property.*

Proof: Suppose an SCF $f : \mathbb{P}^n \to A$ is LOBIC with respect to some belief system μ_N. Since f is LOBIC, we assume that $\mu_i(P_{-i} \mid P_i) > 0$ for all $P_i \in \mathbb{P}$, all $P_{-i} \in \mathbb{P}^{n-1}$ and all $i \in N$. We show that f satisfies the sequential OND property, that is, for all $i \in N$, all $P_i, P_i' \in \mathbb{P}$, and all $P_{-i} \in \mathbb{P}^{n-1}$ with $f(P_i', P_{-i})P_if(P_i, P_{-i})$, there exists an OND sequence $(P^1_{-i}, \ldots, P^k_{-i})$ for f with respect to (P_i, P_i') such that $f(P_i, P_{-i})R_if(P_i', P^k_{-i})$, $f(P_i, P^1_{-i})R_if(P_i', P_{-i})$ and $f(P_i', P_{-i})P_if(P_i', P^1_{-i})$.

Since f is LOBIC, for all agents $i \in N$, all preferences P_i of agent i, all misreported preferences P_i' and all $l = 1, \ldots, m$, we have

$$\sum_{P_{-i} \mid f(P_i, P_{-i}) \in B_l(P_i)} \mu(P_{-i} \mid P_i) \geq \sum_{P_{-i} \mid f(P_i', P_{-i}) \in B_l(P_i)} \mu(P_{-i} \mid P_i). \tag{2.4}$$

Consider an agent $i \in N$, two preferences $\bar{P}_i, \bar{P}_i' \in \mathbb{P}$, and a preference profile $\bar{P}_{-i} \in \mathbb{P}^{n-1}$ of the other agents such that $f(\bar{P}_i', \bar{P}_{-i})\bar{P}_if(\bar{P}_i, \bar{P}_{-i})$. If there does not exist any such instance, then f satisfies sequential OND vacuously. Let $f(\bar{P}_i, \bar{P}_{-i}) = a$ and $f(\bar{P}_i', \bar{P}_{-i}) = b$. We proceed to show that there is an OND sequence $(P^1_{-i}, \ldots, P^k_{-i})$ for f with respect to $(\bar{P}_i, \bar{P}_i')$ such that $f(\bar{P}_i, \bar{P}_{-i})\bar{R}_if(\bar{P}_i', P^k_{-i})$, $f(\bar{P}_i, P^1_{-i})\bar{R}_if(\bar{P}_i', \bar{P}_{-i})$, and $f(\bar{P}_i', \bar{P}_{-i})\bar{P}_if(\bar{P}_i', P^1_{-i})$.

Consider the upper contour set $B(b, \bar{P}_i)$ of b at $\bar{P}_i$. Because $b\bar{P}_ia$, we have $a \notin B(b, \bar{P}_i)$. Applying (3.2.2) to the upper contour set $B(b, \bar{P}_i)$, we have

$$\sum_{P_{-i} \mid f(\bar{P}_i, P_{-i}) \in B(b, \bar{P}_i)} \mu(P_{-i} \mid \bar{P}_i) \geq \sum_{P_{-i} \mid f(\bar{P}_i', P_{-i}) \in B(b, \bar{P}_i)} \mu(P_{-i} \mid \bar{P}_i). \tag{2.5}$$

Because $f(\bar{P}_i, \bar{P}_{-i}) = a$, $a \notin B(b, \bar{P}_i)$, and $\mu_i(\bar{P}_{-i} \mid \bar{P}_i) > 0$, by (2.5) there must exist $\hat{P}_{-i}$ such that $f(\bar{P}_i, \hat{P}_{-i}) \in B(b, \bar{P}_i)$ and $f(\bar{P}_i', \hat{P}_{-i}) \notin B(b, \bar{P}_i)$. Let us denote this $\hat{P}_{-i}$ by P^1_{-i}.

Since $f(\bar{P}_i, P^1_{-i})\bar{R}_if(\bar{P}_i', \bar{P}_{-i})$ and $f(\bar{P}_i', \bar{P}_{-i})\bar{P}_if(\bar{P}_i', P^1_{-i})$, if $f(\bar{P}_i, \bar{P}_{-i})\bar{R}_if(\bar{P}_i{}', P^1_{-i})$ then the sequence (P^1_{-i}) is an OND sequence for f with respect to $(\bar{P}_i, \bar{P}_i')$ satisfying the requirement of Definition 2.3.2 for the current instance. Suppose instead $f(\bar{P}_i{}', P^1_{-i})\bar{P}_if(\bar{P}_i, \bar{P}_{-i})$. Let $f(\bar{P}_i{}', P^1_{-i}) = c$.

Consider the upper contour set $B(c, \bar{P}_i)$. Applying (3.2.2) to $B(c, \bar{P}_i)$, we have

$$\sum_{P_{-i} \mid f(\bar{P}_i, P_{-i}) \in B(c, \bar{P}_i)} \mu(P_{-i} \mid \bar{P}_i) \geq \sum_{P_{-i} \mid f(\bar{P}_i', P_{-i}) \in B(c, \bar{P}_i)} \mu(P_{-i} \mid \bar{P}_i). \tag{2.6}$$

Because $f(\bar{P}_i{}', P^1_{-i})\bar{P}_i f(\bar{P}_i, \bar{P}_{-i})$, we have that $f(\bar{P}_i, \bar{P}_{-i}) \notin B(c, \bar{P}_i)$. Hence, by (2.6) there must exist P^*_{-i} such that $f(\bar{P}_i, P^*_{-i}) \in B(c, \bar{P}_i)$ and $f(\bar{P}'_i, P^*_{-i}) \notin B(c, \bar{P}_i)$. As before, let us denote that P^*_{-i} by P^2_{-i}. By the definition of P^2_{-i}, we have $f(\bar{P}'_i, P^2_{-i}) \notin B(c, \bar{P}_i)$. This, together with the fact that $f(\bar{P}'_i, P^1_{-i}) = c$, implies $f(\bar{P}'_i, P^1_{-i})\bar{P}_i f(\bar{P}'_i, P^2_{-i})$, and hence $P^1_{-i} \neq P^2_{-i}$. Since $f(\bar{P}_i, P^1_{-i})\bar{R}_i f(\bar{P}'_i, \bar{P}_{-i})$ and $f(\bar{P}'_i, \bar{P}_{-i})\bar{P}_i f(\bar{P}'_i, P^1_{-i})$, if $f(\bar{P}_i, \bar{P}_{-i})\bar{R}_i f(\bar{P}_i{}', P^2_{-i})$, then (P^1_{-i}, P^2_{-i}) is an OND sequence for f with respect to $(\bar{P}_i, \bar{P}'_i)$ satisfying the requirements of Definition 2.3.2 for the current instance. If not, then we proceed to the next step.

Continuing in this manner we can construct an OND sequence $(P^1_{-i}, P^2_{-i}, \ldots, P^k_{-i})$ for f with respect to $(\bar{P}_i, \bar{P}'_i)$ such that $f(\bar{P}_i, \bar{P}_{-i})\bar{R}_i f(\bar{P}'_i, P^k_{-i})$, $f(\bar{P}_i, P^1_{-i})\bar{R}_i f(\bar{P}'_i, \bar{P}_{-i})$ and $f(\bar{P}'_i, \bar{P}_{-i})\bar{P}_i f(\bar{P}'_i, P^1_{-i})$. The termination of the process is guaranteed by the fact that $P^1_{-i}, P^2_{-i}, \ldots, P^k_{-i}$ are all distinct. To see why they are distinct, note that, in a similar way as we have shown $f(\bar{P}'_i, P^1_{-i})\bar{P}_i f(\bar{P}'_i, P^2_{-i})$ in the preceding paragraph, we can show $f(\bar{P}'_i, P^1_{-i})\bar{P}_i f(\bar{P}'_i, P^2_{-i})\bar{P}_i \ldots \bar{P}_i f(\bar{P}'_i, P^k_{-i})$. This in particular means $P^1_{-i}, P^2_{-i}, \ldots, P^k_{-i}$ are all distinct.

■

Appendix

2.4 Proof of claim 2.2.2

Proof:

We have shown in Claim 2.2.1 that both μ_1 and μ_2 satisfy TS correlation. So, we need to show that $\hat{f}$ is LOBIC with respect to (μ_1, μ_2). Let $\hat{f}^{\mu_N}_B(P'_i|P_i) = \sum_{P_{-i}|\hat{f}(P'_i, P_{-i}) \in B} \mu_i(P_{-i} \mid P_i)$ denote the aggregate probability induced by $\hat{f}$ according to μ_N on an upper contour set B of P_i when his sincere preference is P_i and he (mis)reports it as P'_i. Therefore, to show that $\hat{f}$ is OBIC with respect to a TS correlated μ_N, we need to show that for each agent $i \in \{1, 2\}$, each sincere preference P_i and each misreport P'_i of agent i, and each upper contour set B of P_i, $\sum_{P_{-i}|\hat{f}(P_i, P_{-i}) \in B} \mu_i(P_{-i} \mid P_i) \geq \sum_{P_{-i}|\hat{f}(P'_i, P_{-i}) \in B} \mu_i(P_{-i} \mid P_i)$, i.e. $\hat{f}^{\mu_N}_B(P_i|P_i) \geq \hat{f}^{\mu_N}_B(P'_i|P_i)$. Suppose that the sincere preference of agent 1 is $P_1 = abc$. Suppose further that he considers a misreport as $P'_1 = acb$. Note that his conditional belief $\mu_1(. \mid abc)$ at $P_1 = abc$ is given in the first row of Table 2.2.2. Further note that the (non-trivial) upper contour sets of the preference abc are $\{a\}$ and $\{a, b\}$. The believed (through $\mu_1(. \mid abc)$) probability that the outcome lies in the upper contour set $\{a\}$ (that is, the outcome is a) when 1 reports abc is

$\hat{f}^{\mu_N}_{\{a\}}(abc|abc) = \mu_1(abc \mid abc) + \mu_1(acb \mid abc) + \mu_1(bca \mid abc) = 0.51 + 0.02 + 0.17 = 0.70$. Similarly, the believed (through $\mu_1(. \mid abc)$, and not through $\mu_1(. \mid acb)$) probability that the outcome is a when 1 misreports his preference as acb is $\hat{f}^{\mu_N}_{\{a\}}(acb|abc) = \mu_1(abc \mid abc) + \mu_1(acb \mid abc) + \mu_1(bac \mid abc) + \mu_1(cba \mid abc) = 0.51 + 0.02 + 0.04 + 0.06 = 0.63$. Since $\hat{f}^{\mu_N}_{\{a\}}(abc|abc) \geq \hat{f}^{\mu_N}_{\{a\}}(acb|abc)$, we have that the requirement of OBIC is satisfied for this instance.

We show that the requirement of OBIC is satisfied for every instance by means of the following tables (Table 2.4.1a - Table 2.4.6b). Each table stands for a sincere preference of an agent, for instance the first table is for $P_1 = abc$. The possible misreports (and the sincere one) are listed in the first column and the corresponding total aggregate probability for different upper contour sets are mentioned in the next columns. Here, for $k = 1, 2$, by B_k we denote the upper contour set of the corresponding sincere preference containing k elements. For instance, in the first table, $B_1 = \{a\}$ and $B_2 = \{a, b\}$. Note that B_3 is not needed to be considered as it contains all the elements a, b, c and hence its aggregate probability will always be one. To help the reader, we have marked the row corresponding to the sincere preference with blue in each table. In each table, the fact that each probability in the first row is weakly bigger than those in the corresponding column establishes that $\hat{f}$ is OBIC.

It remains to show that $\hat{f}$ is LOBIC with respect to μ_N, that is, there is a neighborhood of μ_N such that $\hat{f}$ is OBIC with respect to each $\hat{\mu}_N$ in the neighborhood. Using the continuity of the expectation and the fact that there are finitely many upper contour sets, we can always find a neighborhood of μ_N such that for all $\hat{\mu}_N$ in the neighborhood, all $i \in N$, all P_i and P'_i, and all upper contour set B of P_i, $\hat{f}^{\mu_N}_B(P_i|P_i) > \hat{f}^{\mu_N}_B(P'_i|P_i)$ implies $\hat{f}^{\hat{\mu}_N}_B(P_i|P_i) > \hat{f}^{\hat{\mu}_N}_B(P'_i|P_i)$. However, this does not complete the proof as we have $\hat{f}^{\mu_N}_B(P_i|P_i) = \hat{f}^{\mu_N}_B(P'_i|P_i)$ for some $i \in N$, some P_i, P'_i, and some upper contour sets B of P_i. We have marked such instances in the tables with red color. We need to argue that we can find some neighborhood of μ_N such that for each $\hat{\mu}_N$ in the neighborhood, $\hat{f}^{\hat{\mu}_N}_B(P_i|P_i) \geq \hat{f}^{\hat{\mu}_N}_B(P'_i|P_i)$ for each of these instances. Consider Table **??**. Observe that $\hat{f}^{\mu_N}_{\{a,c\}}(cab|cab) = \hat{f}^{\mu_N}_{\{a,c\}}(cba|cba)$. Note in Table 2.2.1 that for each P_2, $\hat{f}_{\{a,c\}}(cab, P_2) = \hat{f}_{\{a,c\}}(cba, P_2)$. Here, by $\hat{f}_{\{a,c\}}(cab, P_2)$ we denote the probability that the outcome $\hat{f}(cab, P_2)$ belongs to the set $\{a, c\}$ (that is, $\hat{f}_{\{a,c\}}(cab, P_2) = 1$ if $\hat{f}(cab, P_2) \in \{a, c\}$, and $\hat{f}_{\{a,c\}}(cab, P_2) = 0$ otherwise). This fact implies that no matter what the prior belief $\hat{\mu}_N$ is, we will always have $\hat{f}^{\hat{\mu}_N}_{\{a,c\}}(cab|cab) = \hat{f}^{\hat{\mu}_N}_{\{a,c\}}(cba|cba)$. The same logic holds for other instances where $\hat{f}^{\mu_N}_B(P_i|P_i) = \hat{f}^{\mu_N}_B(P'_i|P_i)$

for some B. This proves that $\hat{f}$ is LOBIC with respect to μ_N.

P_1'	$\hat{f}_{B_1}^{\mu_N}(P_1'\|P_1)$	$\hat{f}_{B_2}^{\mu_N}(P_1'\|P_1)$
abc	0.70	0.96
acb	0.63	0.80
bac	0.22	0.94
bca	0.20	0.43
cab	0.53	0.57
cba	0.68	0.72

(a) Upper contour probabilities when $P_1 = abc$

P_1'	$\hat{f}_{B_1}^{\mu_N}(P_1'\|P_1)$	$\hat{f}_{B_2}^{\mu_N}(P_1'\|P_1)$
acb	0.90	0.99
abc	0.54	0.74
bac	0.60	0.77
bca	0.09	0.28
cab	0.53	0.80
cba	0.03	0.80

(b) Upper contour probabilities when $P_1 = acb$

P_1'	$\hat{f}_{B_1}^{\mu_N}(P_1'\|P_1)$	$\hat{f}_{B_2}^{\mu_N}(P_1'\|P_1)$
bac	0.70	0.99
abc	0.15	0.49
acb	0.02	0.86
bca	0.68	0.82
cab	0.51	0.83
cba	0.51	0.70

(a) Upper contour probabilities when $P_1 = bac$

P_1'	$\hat{f}_{B_1}^{\mu_N}(P_1'\|P_1)$	$\hat{f}_{B_2}^{\mu_N}(P_1'\|P_1)$
bca	0.70	0.99
abc	0.15	0.17
acb	0.51	0.52
bac	0.68	0.82
cab	0.02	0.68
cba	0.02	0.34

(b) Upper contour probabilities when $P_1 = bca$

P_1'	$\hat{f}_{B_1}^{\mu_N}(P_1'\|P_1)$	$\hat{f}_{B_2}^{\mu_N}(P_1'\|P_1)$
cab	0.70	0.99
abc	0.01	0.47
acb	0.51	0.83
bac	0.02	0.67
bca	0.17	0.68
cba	0.67	0.99

(a) Upper contour probabilities when $P_1 = cab$

P_1'	$\hat{f}_{B_1}^{\mu_N}(P_1'\|P_1)$	$\hat{f}_{B_2}^{\mu_N}(P_1'\|P_1)$
cba	0.75	0.98
abc	0.23	0.77
acb	0.03	0.04
bac	0.51	0.76
bca	0.52	0.97
cab	0.55	0.78

(b) Upper contour probabilities when $P_1 = cba$

P_2'	$\hat{f}^{\mu_N}_{B_1}(P_2'\|P_2)$	$\hat{f}^{\mu_N}_{B_2}(P_2'\|P_2)$
abc	0.90	0.99
acb	0.79	0.80
bac	0.02	0.49
bca	0.71	0.83
cab	0.10	0.61
cba	0.02	0.53

(a) Upper contour probabilities when $P_2 = abc$

P_2'	$\hat{f}^{\mu_N}_{B_1}(P_2'\|P_2)$	$\hat{f}^{\mu_N}_{B_2}(P_2'\|P_2)$
acb	0.90	0.99
abc	0.82	0.83
bac	0.51	0.53
bca	0.11	0.31
cab	0.18	0.98
cba	0.51	0.98

(b) Upper contour probabilities when $P_2 = acb$

P_2'	$\hat{f}^{\mu_N}_{B_1}(P_2'\|P_2)$	$\hat{f}^{\mu_N}_{B_2}(P_2'\|P_2)$
bac	0.90	0.99
abc	0.51	0.98
acb	0.02	0.80
bca	0.62	0.83
cab	0.01	0.54
cba	0.01	0.10

(a) Upper contour probabilities when $P_2 = bac$

P_2'	$\hat{f}^{\mu_N}_{B_1}(P_2'\|P_2)$	$\hat{f}^{\mu_N}_{B_2}(P_2'\|P_2)$
bca	0.78	0.98
abc	0.02	0.53
acb	0.51	0.52
bac	0.74	0.75
cab	0.01	0.47
cba	0.01	0.75

(b) Upper contour probabilities when $P_2 = bca$

P_2'	$\hat{f}^{\mu_N}_{B_1}(P_2'\|P_2)$	$\hat{f}^{\mu_N}_{B_2}(P_2'\|P_2)$
cab	0.78	0.99
abc	0.06	0.85
acb	0.02	0.94
bac	0.01	0.26
bca	0.51	0.54
cba	0.74	0.99

(a) Upper contour probabilities when $P_2 = cab$

P_2'	$\hat{f}^{\mu_N}_{B_1}(P_2'\|P_2)$	$\hat{f}^{\mu_N}_{B_2}(P_2'\|P_2)$
cba	0.90	0.99
abc	0.20	0.37
acb	0.51	0.71
bac	0.09	0.99
bca	0.02	0.40
cab	0.54	0.63

(b) Upper contour probabilities when $P_2 = cba$

P'_1	(P_{-1}, P'_{-1})	$(\bar{P}_{-1}, \bar{P}'_{-1})$
acb	(bac,-)	(cba,bca)
bac	(bac,cba)	(cab,bca)
bca	(bac,cba)	(cab,bca)
cab	(bac,cba)	
cba	(bac,cba)	

(a) $P_1 = abc$

P'_1	(P_{-1}, P'_{-1})	$(\bar{P}_{-1}, \bar{P}'_{-1})$
abc	(cba,acb)	(cab,acb)
acb	(cab,abc)	
bac	(abc,acb)	
cab	(cab,abc)	
cba	(cab,abc)	

(b) $P_1 = bca$

P'_2	(P_{-2}, P'_{-2})	$(\bar{P}_{-2}, \bar{P}'_{-2})$
acb	(bac,cba)	(bca,cba)
bac	(bca,abc)	
bca	(bca,cab)	
cab	(bac,cba)	(bca,cba)
cba		

(a) $P_2 = abc$

P'_2	(P_{-2}, P'_{-2})	$(\bar{P}_{-2}, \bar{P}'_{-2})$
abc	(abc,bca)	
acb	(abc,cba)	
bca	(abc,cab)	(acb,cba)
cab	(abc,cab)	
cba	(abc,bac)	

(b) $P_2 = bac$

■

2.5 Proof of Claim 2.3.1

Proof: For every preference P_i of agent $i \in \{1, 2\}$, we have a table in Tables 2.4.7a - **??**. In the corresponding table, the first column presents preferences P'_i via which agent i can manipulate. Every other column presents a pair (P_{-i}, P'_{-i}) [or $(\bar{P}_{-i}, \bar{P}'_{-i})$ or $(\hat{P}_{-i}, \hat{P}'_{-i})$] such that agent i manipulates at (P_i, P_{-i}) [or $(P_i, \bar{P}_{-i})$ or $(P_i, \hat{P}_{-i})$] via P'_i and P'_{-i} [or $\bar{P}'_{-i}$ or $\hat{P}'_{-i}$] satisfies the conditions in the definition of OND (Definition 2.2.5). For instance, Table 2.4.7a considers manipulation by agent 1 when his sincere preference is *abc*. The first element $P'_1 = acb$ and the second element $(P_{-1}, P'_{-1}) = (bac, -)$ in the first row indicates the fact that agent 1 manipulates at the preference profile (abc, bac) via acb , and there is no preference of agent 2 where the conditions in the Definition 2.2.5 are satisfied. The third element $(\bar{P}_{-1}, \bar{P}'_{-1}) = (cba, bca)$ of the first row indicates that the agent 1 manipulates at the preference profile (abc, cba) via acb and the conditions in Definition 2.2.5 are satisfied at the preference bca of agent 2.

It follows from the Tables 2.4.7a - 2.5.6b that $\hat{f}$ satisfies the OND property for all cases except for the case when $P_1 = abc, P_1' = acb$ and $P_{-1} = bac$ (i.e. $P_2 = bac$).

P_1'	(P_{-1}, P_{-1}')	$(\bar{P}_{-1}, \bar{P}_{-1}')$
acb	(bac, –)	(cba, bca)
bac	(bac, cba)	(cab, bca)
bca	(bac, cba)	(cab, bca)
cab	(bac, cba)	
cba	(bac, cba)	

(a) $P_1 = abc$

P_1'	(P_{-1}, P_{-1}')
abc	(bca, cba)
bac	(cab, cba)
bca	(cab, cba)
cab	(bca, bac)
cba	(bca, bac)

(b) $P_1 = acb$

P_1'	(P_{-1}, P_{-1}')	$(\bar{P}_{-1}, \bar{P}_{-1}')$
abc	(cab, bac)	(cba, bac)
acb	(cba, cab)	
bca	(acb, abc)	
cab		
cba		

(a) $P_1 = bac$

P_1'	(P_{-1}, P_{-1}')	$(\bar{P}_{-1}, \bar{P}_{-1}')$
abc	(cba, acb)	(cab, acb)
acb	(cab, abc)	
bac	(abc, acb)	
cab	(cab, abc)	
cba	(cab, abc)	

(b) $P_1 = bca$

P_1'	(P_{-1}, P_{-1}')
abc	(bac, cab)
acb	(bac, bca)
bac	
bca	(abc, cab)
cba	(acb, bca)

(a) $P_1 = cab$

P_1'	(P_{-1}, P_{-1}')	$(\bar{P}_{-1}, \bar{P}_{-1}')$
abc	(bac, cab)	
acb	(bca, cba)	
bac	(abc, cab)	(bca, cab)
bca	(abc, cab)	(bca, cab)
cab	(bca, acb)	

(b) $P_1 = cba$

P_2'	(P_{-2}, P'_{-2})	$(\bar{P}_{-2}, \bar{P}'_{-2})$
acb	(bac, cba)	(bca, cba)
bac	(bca, abc)	
bca	(bca, cab)	
cab	(bac, cba)	(bca, cba)
cba		

(a) $P_2 = abc$

P_2'	(P_{-2}, P'_{-2})	$(\bar{P}_{-2}, \bar{P}'_{-2})$
abc	(bca, bac)	(cba, bac)
bac		
bca	(cba, cab)	
cab	(bca, abc)	
cba	(bca, abc)	

(b) $P_2 = acb$

P_2'	(P_{-2}, P'_{-2})	$(\bar{P}_{-2}, \bar{P}'_{-2})$
abc	(abc, bca)	
acb	(abc, cba)	
bca	(abc, cab)	(acb, cba)
cab	(abc, cab)	
cba	(abc, bac)	

(a) $P_2 = bac$

P_2'	(P_{-2}, P'_{-2})	$(\bar{P}_{-2}, \bar{P}'_{-2})$	$(\hat{P}_{-2}, \hat{P}'_{-2})$
abc			
acb	(cba, acb)		
bac	(abc, acb)	(cab, acb)	(cba, acb)
cab	(abc, bca)	(cba, bca)	
cba	(abc, acb)	(cba, acb)	

(b) $P_2 = bca$

P_2'	(P_{-2}, P'_{-2})	$(\bar{P}_{-2}, \bar{P}'_{-2})$
abc	(abc, bac)	(bca, cab)
acb	(abc, bca)	
bac	(abc, cab)	
bca	(abc, acb)	
cba	(bac, acb)	(bca, acb)

(a) $P_2 = cab$

P_2'	(P_{-2}, P'_{-2})
abc	
acb	
bac	(abc, bac)
bca	(acb, abc)
cab	(acb, bca)

(b) $P_2 = cba$

∎

3

Ordinal Bayesian incentive-compatible voting rules with correlated belief under betweenness property

3.1 Introduction

A social choice function (SCF) is called ordinal Bayesian incentive compatible (OBIC) with respect to a prior (belief) if no agent can increase his expected utility (according to his prior) by misreporting his sincere preference. An SCF is called locally robust OBIC (LOBIC) with respect to a prior if it is OBIC with respect to each prior in a small neighborhood of the original prior. A LOBIC SCF incentivizes agents to reveal their sincere preferences even if the designer is *slightly* unsure about their beliefs. The objective of this paper is to explore the structure of SCFs that are LOBIC with respect to correlated priors.

The structure of OBIC SCFs when priors are independent is well-explored in the literature (see [16], [17] and [12] for details). In contrast to the case of independent priors, to the best of our knowledge the structure of LOBIC SCFs with respect to correlated priors is relatively less explored. Only paper we know in this topic is [5].

[5] define positive correlation through the notion of top-set. Top-set (TS) correlation ensures that for every agent i and at every preference P_i, the probability (according to the conditional distribution) of other agents' top k alternatives (for any k) agreeing with agent i's own top k alternatives is greater than the probability of the event where other agents' top k alternatives is different than that of agent i. [1]

In this paper, we consider a different measure for correlation. We explain the reason in the following. Suppose that there are three alternatives a, b, and c and two agents 1 and 2. Suppose further that agent 1 has preference $P_1 = abc$. If agent 1 believes that agent 2 is of his type (that is, positively correlated with him), then he must deem preferences that are "closer" to abc more likely for 2 than the ones that are "farther away" from abc. For instance, $\mu_1(bac \mid P_1)$ and $\mu_1(acb \mid P_1)$ should be bigger than $\mu_1(cba \mid P_1)$. However, this is not ensured by the definition of top-set correlation. For instance, beliefs such as $\mu_1(abc \mid P_1) = 0.5 + \varepsilon$ and $\mu_1(cba \mid P_1) = 0.5 - \varepsilon$, where $\varepsilon > 0$ is arbitrarily small, satisfy top-set correlation but do not really represent positive correlation.[2]. However, according to such a belief, any preference which is obtained through a small change in P_1 gets much lower probability (in fact zero probability) than the preference that is completely opposite of P_1. We intend to introduce a notion for measuring positive correlation that takes care of the fact that farther away preferences are believed to be less likely than the closer ones.

We use the notion of 'betweenness' to formulate the correlation structure of the priors. For an illustration, suppose that there are two agents 1 and 2, and three alternatives a, b and c. Suppose further that agent 1 has the preference abc. Note that the preference bac differs from abc by the swap of a and b, whereas the preference bca differs from abc by the swap of both a and b, and a and c. In some sense, the preference bca is "farther away" from abc than bac is from abc. In such situations, we say that the preference bac lies between abc and bca. Now, suppose that agent 1 believes that agent 2 is of the same type, that is, has a preference that is similar to his preference. Positive correlation under betweenness relation says that agent 1 will deem the preference abc more likely than the preference bac, and the preference bac more likely than the preference bca. On the other hand, if agent 1 believes that agent 2 is of his opposite type, then negative correlation under betweenness relation imposes exactly opposite

[1] See Definition 10 in [5] for a rigorous description.

[2] The opposite preference P' of a preference P is the one that reverses the ordering of the alternatives in P, that is, for all alternatives a and b, aPb if and only if $bP'a$.

structure on his belief about agent 2's preference. For an agent, we say a profile of other agents gets closer to his preference if the preference of each positively correlated agent gets closer and that of each negatively correlated agent gets farther away (with respect to the betweenness relation).

We assume the coexistence of both positively correlated and negatively correlated beliefs. More formally, each agent classifies other agents into two categories: the positively correlated agents and the negatively correlated agents, and forms his belief accordingly. We introduce the notion of strong OND property which, roughly, says that (i) if the outcome for an agent improves by a unilateral deviation of that agent at a given profile of other agents, then there will be another profile of other agents where the same unilateral deviation of that agent will cause at least that much disimprovement,[3] and (ii) the outcome weakly improves for an agent as the profile of other agents gets 'closer' (as defined in the previous paragraph) to his preference. In this paper, we prove that the strong OND property is sufficient to ensure that an SCF is LOBIC with respect to some correlated under betweenness (CBR) prior.

The rest of the paper is organized as follows. Section 3.2 introduces the basic model, and Section 3.3 presents the result and its proof.

3.2 Preliminaries

Let $N = \{1, 2, \ldots, n\}$ be a set of agents and A be a set of alternatives with $|A| = m$. A preference P is defined as a complete, transitive, and antisymmetric binary relation on A. We denote the weak part of a preference P by R, that is, for two alternatives a and b, aRb means either aPb or $a = b$.

Each agent $i \in N$ has a domain of (admissible) preferences $\mathcal{D}_i$ which is subset of the set of all preferences on A. We denote by $\mathcal{D}_N$ the product set $\prod_{i\in N} \mathcal{D}_i$, and for $i \in N$, we denote by $\mathcal{D}_{-i}$ the set $\prod_{j\neq i} \mathcal{D}_j$. An element $P_N = (P_1, \ldots, P_n)$ of $\mathcal{D}_N$ is called a preference profile. For a preference profile P_N, we denote the restriction of P_N to $N \setminus \{i\}$ by P_{-i}.

A belief μ_i of an agent i is a probability distribution on $\mathcal{D}_N$. A collection $\mu_N = (\mu_1, \ldots, \mu_n)$ of beliefs is called a prior (profile). For a prior μ_i and a preference P_i of agent i, we denote by $\mu_i(. \mid P_i)$ the conditional belief (conditional probability distribution on $\mathcal{D}_{-i}$ given P_i) of i given the preference P_i.

[3]For instance, if a unilateral deviation of an agent improves the rank of the outcome at the sincere preference of the agent, say, from ten to five, at a profile of other agents, then there will be another profile of other agents where the same deviation will decrease the rank of the outcome, say, from three (that is, smaller or equal to five) to eleven (that is, higher or equal to ten).

A utility function $u : A \to \mathbb{R}$ is said to represent a preference P if for all $a, b \in A$, we have aPb if and only if $u(a) > u(b)$.

A social choice function (SCF) (on $\mathcal{D}_N$) is a mapping $f : \mathcal{D}_N \to A$.

Definition 3.2.1 *An SCF f is **ordinal Bayesian incentive compatible (OBIC)** with respect to a prior μ_N if for all $i \in N$, all $P_i, P'_i \in \mathcal{D}_i$, and all utility functions u_i representing P_i, we have*

$$\sum_{P_{-i} \in \mathcal{D}_{-i}} u_i(f(P_i, P_{-i}))\mu_i(P_{-i} \mid P_i) \geq \sum_{P_{-i} \in \mathcal{D}_{-i}} u_i(f(P'_i, P_{-i}))\mu_i(P_{-i} \mid P_i).$$

Definition 3.2.2 *An SCF f is **locally robust OBIC (LOBIC)** with respect to a prior μ_N if there exists $\varepsilon > 0$ such that f is OBIC with respect to all priors in $\bar{\mu}_N \in B_\varepsilon(\mu_N)$.*[4]

As mentioned earlier,our notion of correlation is based on the notion of betweenness. A preference P is said to lie between two preferences P^1 and P^2, denoted by $P \in (P^1, P^2)$, if $P^1 \Delta P \subseteq P^1 \Delta P^2$, where $P \Delta P' = \{\{x, y\} \mid xPy \text{ and } yP'x\}$ denotes the set of (unordered) pairs of alternatives whose relative orderings are different in P and P'. In other words, P lies between P^1 and P^2 if P is "more similar" to P^1 than P^2 is to P^1.

We assume that for every agent $i \in N$, there exists a *fixed* (does not depend on the preferences of i) partition $\{N_i^+, N_i^-\}$ of $N \setminus \{i\}$ such that agent i has a positively correlated belief about agents in N_i^+, and a negative correlated belief about the agents in N_i^-. We call such a prior (profile) a correlated prior (profile) under betweenness relation and denote the set of all such correlated priors by $\mathcal{M}$. Below, we provide a formal definition of this. For ease of presentation, we write $P'_{-i} \in \langle P_i, P_{-i} \rangle$ to mean $P'_j \in (P_i, P_j)$ for all $j \in N_i^+$ and $P_j \in (P_i, P'_j)$ for all $j \in N_i^-$.

Definition 3.2.3 *A prior μ_i of an agent $i \in N$ is **correlated under betweenness relation (CBR)** if for all $P_i \in \mathcal{D}_i$ and all $P_{-i}, P'_{-i} \in \mathcal{D}_{-i}$ with the property that $P'_{-i} \in \langle P_i, P_{-i} \rangle$, we have $\mu_i(P'_{-i}|P_i) > \mu_i(P_{-i}|P_i)$. A prior (profile) μ_N is correlated if μ_i is correlated for each $i \in N$.*

[4]We denote by $B_\varepsilon(\mu_N)$ the open ball of radius ε centered at μ_N, that is, the set of all priors $\bar{\mu}_N$ that are at most ε distance from μ_N.

Definition 3.2.4 *An SCF $f : \mathcal{D}_N \to A$ satisfies the **ordinal nondomination (OND)** property if for all $i \in N$, all $P_i, P'_i \in \mathcal{D}_i$, and all $P_{-i} \in \mathcal{D}_{-i}$ such that $f(P'_i, P_{-i})P_i f(P_i, P_{-i})$, there exists $P'_{-i} \in \mathcal{D}_{-i}$ with the property that $f(P_i, P'_{-i})R_i f(P'_i, P_{-i})$ and $f(P_i, P_{-i})R_i f(P'_i, P'_{-i})$.*

Definition 3.2.5 *An SCF $f : \mathcal{D}_N \to A$ satisfies the **strong OND property** if f satisfies the OND property and for all $i \in N$, all $P_i \in \mathcal{D}_i$, and all $P_{-i}, P'_{-i} \in \mathcal{D}_{-i}$ such $P_{-i} \in \langle P_i, P'_{-i}\rangle$, we have $f(P_i, P_{-i})R_i f(P_i, P'_{-i})$.*

3.3 Theorem

Theorem 3.3.1 *Suppose that an SCF satisfies the strong OND property. Then, it is LOBIC with respect to some CBR prior.*

Proof: Suppose an SCF $f : \mathcal{D}_N \to A$ satisfies strong OND. We show that there exists a CBR prior μ_N such that f is LOBIC with respect to μ_N. We proceed to construct the mentioned CBR prior μ_N. Let μ_N be such that

(i) $\mu_i(P_{-i}|P_i) > 0$ for all $i \in N$, all $P_i \in \mathcal{D}_i$, and all $P_{-i} \in \mathcal{D}_{-i}$

(ii) $\mu_i(P_{-i}|P_i) > \mu_i(P'_{-i}|P_i)$ for all $i \in N$, all $P_{-i} \in \mathcal{D}_i$ such that $P_{-i} \in \langle P_i, P'_{-i}\rangle$,

(iii) $\mu_i(P_{-i}|P_i) > \sum_{P'_{-i}|f(P_i,P_{-i})P_i f(P_i,P'_{-i})} \mu_i(P'_{-i}|P_i)$ for all $i \in N$ and all $P_i \in \mathcal{D}_i$.

Here, (i) is a requirement for locally robustness, (ii) ensures that μ_N is a CBR prior, and (iii) is a technical condition that we need to ensure that f is LOBIC with respect to μ_N.

Since f satisfies the strong OND property, for all $i \in N$, all $P_i \in \mathcal{D}_i$, and all $P_{-i}, P'_{-i} \in \mathcal{D}_{-i}$ such $P_{-i} \in \langle P_i, P'_{-i}\rangle$, we have $f(P_i, P_{-i})R_i f(P_i, P'_{-i})$. This implies that for any given P_{-i}, there is no P''_{-i} in the set $\{P'_{-i} \mid f(P_i, P_{-i})P_i f(P_i, P'_{-i})\}$ such that $P''_{-i} \in \langle P_i, P_{-i}\rangle$. Hence, the prior μ_N is indeed a CBR prior. We proceed to show that f is LOBIC with respect μ_N.

Consider (arbitrary) $i \in N$, $P_i, P'_i \in \mathcal{D}_i$ and a utility representation u_i of P_i. Let $M(P_i, P'_i) = \{\bar{P}_{-i} \in \mathcal{D}_{-i} \mid f(P'_i, \bar{P}_{-i})P_i f(P_i, \bar{P}_{-i})\}$ be the set of all preference profiles of the agents in $N \setminus i$ such that agent i can manipulate by misreporting his sincere preference P_i as P'_i. Furthermore, let $M(P_i, P'_i) = \{P_{-i} \in \mathcal{D}_{-i} \mid f(P_i, P_{-i})P_i f(P'_i, P_{-i})\}$ be the set of all preference profiles of the agents in $N \setminus i$

such that agent i cannot manipulate by misreporting his sincere preference P_i as P_i'. Clearly, $M(P_i, P_i') \cap M(P_i, P_i') = \emptyset$. To show that f is LOBIC, we need to show that

$$\begin{aligned} &\textstyle\sum_{P_{-i} \in M(P_i, P_i')} \mu_i(P_{-i}|P_i)[u_i\big(f(P_i, P_{-i})\big) - u_i\big(f(P_i', P_{-i})\big)] \\ &\geq \textstyle\sum_{\bar{P}_{-i} \in M(P_i, P_i')} \mu_i(\bar{P}_{-i}|P_i)[u_i\big(f(P_i', \bar{P}_{-i})\big) - u_i\big(f(P_i, \bar{P}_{-i})\big)]. \end{aligned} \tag{3.1}$$

For each $P_{-i} \in M(P_i, P_i')$, we define

$$S(P_{-i}) = \{\bar{P}_{-i} \in M(P_i, P_i') \mid f(P_i, P_{-i}) R_i f(P_i', \bar{P}_{-i}) \text{ and } f(P_i, \bar{P}_{-i}) R_i f(P_i', P_{-i})\}.$$

Since f satisfies the strong OND property (and hence, the OND property in particular), we obtain the following two facts.

(i) For all $P_{-i} \in M(P_i, P_i')$ with $S(P_{-i}) \neq \emptyset$ and all $\bar{P}_{-i} \in S(P_{-i})$, we have

$$u_i\big(f(P_i, P_{-i})\big) - u_i\big(f(P_i', P_{-i})\big) \geq u_i\big(f(P_i', \bar{P}_{-i})\big) - u_i\big(f(P_i, \bar{P}_{-i})\big),$$

which means

$$u_i\big(f(P_i, P_{-i})\big) - u_i\big(f(P_i', P_{-i})\big) \geq \max_{\bar{P}_{-i} \in S(P_{-i})} u_i\big(f(P_i', \bar{P}_{-i})\big) - u_i\big(f(P_i, \bar{P}_{-i})\big). \tag{3.2}$$

(ii)

$$\cup_{P_{-i} \in M(P_i, P_i')} S(P_{-i}) = M(P_i, P_i'). \tag{3.3}$$

Consider $P_{-i} \in M(P_i, P_i')$. For every $\bar{P}_{-i} \in S(P_{-i})$, we have $f(P_i, P_{-i}) P_i f(P_i, \bar{P}_{-i})$. This, together with Part (iii) of the definition of μ_N, gives us

$$\mu_i(P_{-i}|P_i) > \sum_{\bar{P}_{-i} \in S(P_{-i})} \mu_i(\bar{P}_{-i}|P_i). \tag{3.4}$$

Using standard algebra, we have

$$\begin{aligned}
&\textstyle\sum_{\bar{P}_{-i}\in S(P_{-i})} \mu_i(\bar{P}_{-i} \mid P_i)\Big[u_i\big(f(P_i', \bar{P}_{-i})\big) - u_i\big(f(P_i, \bar{P}_{-i})\big)\Big] \\
&\le \textstyle\sum_{\bar{P}_{-i}\in S(P_{-i})} \mu_i(\bar{P}_{-i} \mid P_i) \max_{\bar{P}_{-i}\in S(P_{-i})} \Big[u_i\big(f(P_i', \bar{P}_{-i})\big) - u_i\big(f(P_i, \bar{P}_{-i})\big)\Big] \\
&= \max_{\bar{P}_{-i}\in S(P_{-i})} \Big[u_i\big(f(P_i', \bar{P}_{-i})\big) - u_i\big(f(P_i, \bar{P}_{-i})\big)\Big] \textstyle\sum_{\bar{P}_{-i}\in S(P_{-i})} \mu_i(\bar{P}_{-i} \mid P_i).
\end{aligned} \tag{3.5}$$

By (3.2), we have (3.6), and by (3.4) we have (3.7).

$$\begin{aligned}
&\max_{\bar{P}_{-i}\in S(P_{-i})} \Big[u_i\big(f(P_i', \bar{P}_{-i})\big) - u_i\big(f(P_i, \bar{P}_{-i})\big)\Big] \textstyle\sum_{\bar{P}_{-i}\in S(P_{-i})} \mu_i(\bar{P}_{-i} \mid P_i) \\
&\le \Big[u_i\big(f(P_i, P_{-i})\big) - u_i\big(f(P_i', P_{-i})\big)\Big] \textstyle\sum_{\bar{P}_{-i}\in S(P_{-i})} \mu_i(\bar{P}_{-i} \mid P_i).
\end{aligned} \tag{3.6}$$

$$\begin{aligned}
&\Big[u_i\big(f(P_i, P_{-i})\big) - u_i\big(f(P_i', P_{-i})\big)\Big] \textstyle\sum_{\bar{P}_{-i}\in S(P_{-i})} \mu_i(\bar{P}_{-i} \mid P_i) \\
&\le \Big[u_i\big(f(P_i, P_{-i})\big) - u_i\big(f(P_i', P_{-i})\big)\Big] \mu_i(P_{-i} \mid P_i).
\end{aligned} \tag{3.7}$$

Combining (3.5), (3.6), and (3.7), we obtain (3.8).

$$\Big[u_i\big(f(P_i, P_{-i})\big) - u_i\big(f(P_i', P_{-i})\big)\Big] \mu_i(P_{-i} \mid P_i) \ge \sum_{\bar{P}_{-i}\in S(P_{-i})} \mu_i(\bar{P}_{-i} \mid P_i)\Big[u_i\big(f(P_i', \bar{P}_{-i})\big) - u_i\big(f(P_i, \bar{P}_{-i})\big)\Big]. \tag{3.8}$$

By summing both sides of (3.8) over the elements P_{-i} of $M(P_i, P_i')$, we have

$$\begin{aligned}
&\textstyle\sum_{P_{-i}\in M(P_i,P_i')} \Big[u_i\big(f(P_i, P_{-i})\big) - u_i\big(f(P_i', P_{-i})\big)\Big] \mu_i(P_{-i} \mid P_i) \\
&\ge \textstyle\sum_{P_{-i}\in M(P_i,P_i')} \sum_{\bar{P}_{-i}\in S(P_{-i})} \mu_i(\bar{P}_{-i} \mid P_i)\Big[u_i\big(f(P_i', \bar{P}_{-i})\big) - u_i\big(f(P_i, \bar{P}_{-i})\big)\Big].
\end{aligned} \tag{3.9}$$

Since by (3.3) $\cup_{P_{-i}\in M(P_i,P_i')} S(P_{-i}) = M(P_i, P_i')$, all the elements of $M(P_i, P_i')$ appear in the summand in the R.H.S. of (3.9). Therefore, it follows that

$$
\begin{aligned}
&\textstyle\sum_{P_{-i}\in M(P_i,P_i')}\Big[u_i\big(f(P_i,P_{-i})\big)-u_i\big(f(P_i',P_{-i})\big)\Big]\mu_i(P_{-i}\mid P_i)\\
&\geq \textstyle\sum_{\bar{P}_{-i}\in M(P_i,P_i')}\mu_i(\bar{P}_{-i}\mid P_i)\Big[u_i\big(f(P_i',\bar{P}_{-i})\big)-u_i\big(f(P_i,\bar{P}_{-i})\big)\Big].
\end{aligned}
$$

as required by (3.1). This completes the proof of the theorem.

■

4
Social choice with network-based common knowledge

4.1 Introduction

4.1.1 Description and background of the problem

We study a social choice environment in which agents' preferences are not entirely private information. Instead, there exists *partial common knowledge* about preferences, which is structured through an underlying network. In this model, each agent observes the preferences of their neighbors in the network, and this informational structure is commonly known—not only among the agents but also to the mechanism designer. Nevertheless, as in standard mechanism design settings, the designer does not have direct access to the true preferences of any agent. The central aim of the designer is to exploit this partial information in order to design mechanisms that remain truthful, despite the lack of full information.

Formally, a *social choice function* maps every admissible profile of agents' preferences to a single alternative. Such a function is said to be *strategy-proof* if no agent can benefit by misreporting their own preference, regardless of the reports of others. However, in our framework, mechanisms are richer objects:

each agent is required to report not only their own preferences but also those of their network neighbors. The mechanism then maps each possible profile of such extended reports to an alternative.

A report is said to be *consistent* if no two agents provide conflicting reports about any individual's preference—either their own or someone else's. An inconsistent report signals that at least one agent is misreporting. In this paper, we only consider unilateral misreporting by individual agent, thus ruling out cases when agents may still coordinate their misreports so that no contradictions arise. A mechanism is said to be *incentive compatible* if no agent can benefit by misreporting either their own or others' preferences, when other agents are reporting truthfully.

We define a social choice function to be *incentive-compatible extendable* if there exists an incentive-compatible mechanism whose outcome coincides with the function whenever the agents' reports are consistent. As is well-known from the Gibbard–Satterthwaite theorem and related impossibility results, achieving strategy-proofness along with efficiency and non-dictatorship is infeasible in general domains. This paper seeks to circumvent such impossibilities by leveraging the partial common knowledge of preferences embedded in the network structure.

The central innovation of our approach lies in how the mechanism designer utilizes the partial information embedded in the network structure. Beginning with a fixed social choice function—which prescribes an outcome for every complete profile of admissible preferences—the designer solicits from each agent not only a report of their own preferences, but also their knowledge about the preferences of their immediate neighbors in the network. This reporting structure is justified by the assumption that agents possess reliable local knowledge of their neighbors' preferences, and that such information is mutually known and verifiable within the network.

Once the reports are collected, the mechanism defines outcomes based on the reported preferences, with a crucial addition: it incorporates a system of *penalties* or *sanctions* that are triggered in the event of inconsistencies or suspected misreporting. In particular, the mechanism is constructed such that agents who are suspected of dishonesty—whether by reporting inconsistent information or by deviating from the expected truth-telling behavior—receive outcomes that are strictly less favorable than those they would obtain under truthful and consistent reporting. This discourages manipulation by making deception costly in terms of final allocation.

Importantly, this design paradigm operates entirely within the framework of *individual strategic behavior*. It focuses on ensuring that each agent, when acting in isolation, has no incentive to misreport either their own preferences or those of their neighbors. By doing so, the mechanism preserves incentive compatibility in a decentralized and informationally constrained setting, while circumventing classical impossibility results. One drawback to this method is that, we are not able to capture manipulation when multiple agents coordinate while manipulating.

It is important to note that the structure of incentive-compatible mechanisms is inherently dependent on the underlying network structure among the agents. More precisely, the informational links encoded by the network—indicating which agents have knowledge of others' preferences—play a crucial role in determining the design space and feasibility of truthful mechanisms. Intuitively, the richer the informational environment (i.e., the greater the number of neighbors an agent has), the more leverage the designer possesses to construct mechanisms that deter strategic misreporting. In other words, networks with higher connectivity expand the designer's ability to cross-verify agents' reports, thereby enhancing the scope for implementing incentive-compatible outcomes.

Such informational structures naturally arise from social networks—e.g., families, friendships, community memberships, or institutional affiliations—where agents possess local knowledge about the preferences of others. The existence and relevance of such networks are well-documented in both theoretical and empirical literatures, making our model applicable to a wide range of real-world scenarios.

4.1.2 Our contribution

We investigate the structural properties of social choice functions that can be extended in an incentive-compatible manner, with respect to various network topologies representing partial knowledge of agents' preferences. Specifically, we analyze three distinct cases based on the structure of the underlying network:

(i) **Dense Local Knowledge:** Each agent has at least three neighbors in the network.

(ii) **Minimal Network:** The network consists of exactly two agents connected by a single link.

(iii) **General Connected Networks:** The network is an arbitrary connected graph.

In the first case, where every agent is connected to at least two others, we demonstrate a strong positive result: *every social choice function* is extendable in an incentive-compatible manner. This follows from the fact that the richness in neighborhood information enables the designer to effectively detect and penalize any misreporting, thereby ensuring truthful behavior.

In the second case, where the network is minimal and consists of only two agents with a single link between them, we identify a critical structural condition on the social choice function, which we refer to as the *Lower Contour Set Intersection Property (LCSIP)*. We show that the LCSIP is both *necessary and sufficient* for the incentive-compatible extendability of a social choice function in this minimal network setting.

In the third and most general case, we consider arbitrary connected networks. Here, we extend the intuition from the minimal case by generalizing the LCSIP. We formulate a broader condition that takes into account the global consistency of reported preferences across the entire network. We then establish that this generalized condition is *necessary and sufficient* for the existence of an incentive-compatible mechanism that coincides with the given social choice function on all consistent reports.

These results collectively highlight how the feasibility of incentive-compatible implementation is fundamentally shaped by the informational topology of the network and the alignment properties of the social choice rule.

4.1.3 Existing literature

Various papers in the mechanism design literature incorporate networks in their underlying frameworks, with [10] serving as the seminal reference on this topic. However, to the best of our knowledge, no existing papers model the precise problem addressed in this paper within a similar framework.

Both [11] and [4] investigate network formation problems where the graph structure is endogenous, as agents strategically form and sever links to alter the network topology. Additionally, [18] employs network theory to analyze cooperative structures in games.

The study by [6] closely resembles our model, as they also consider scenarios where agents possess information about their neighbors' preferences, and the mechanism designer aims to exploit this local information for improved mechanism design. Furthermore, their principal finding parallels our result in

Theorem 4.3.1, in the sense that complete information extraction through the network is feasible if and only if the network is sufficiently rich—specifically, each pair of agents must share at least one common neighbor. Nevertheless, their approach significantly differs from ours by assuming that agents have imperfect knowledge regarding their neighbors' private information. One other paper we would like to highlight is [8] in the implementation literature. Although their underlying model lacks explicit graph structures, their approach involves each agent reporting the preferences of every other agent, which the designer subsequently utilizes to formulate the mechanism based on these *reported preferences*.

4.2 Preliminaries

Let $N = \{1, \ldots, n\}$ be the set of agents and $A = \{a, b, \ldots\}$ be the set of alternatives. A (strict) preference P over the alternatives is an anti-symmetric linear order on A. We denote the weak part of a preference P by R, that is $a\,R\,b$ implies either $a\,P\,b$ or $a = b$. Given a preference P, we denote the k-th ranked alternative in P with $r_k(P)$. For any alternative $a \in A$ and preference P, we define the lower contour set of a with respect to P as the set $L(a, P) = \{x \in A \mid aPx \text{ or } a = x\}$. The strict upper contour set of a with respect to P is defined as the set $U(a, P) = \{x \in a \mid xPa\}$. It follows immediately from the definition that, for any alternative a and preference P, the lower contour set and the strict upper contour set are disjoint, that is, $L(a, P) \cap U(a, P) = \emptyset$. We denote by $\mathcal{P}$ the set of all preferences. A preference profile $P_N = (P_1, \ldots, P_n)$ is a collection of preferences, one for each agent in N. The set of all possible preference profiles is denoted by $\mathcal{P}^n$.

The agents are connected through an exogenously given network modeled through an undirected connected graph $G = \langle N, E\rangle$, where $E \subseteq N^2$ is the set of edges in G. Throughout this paper, we assume that $(i, i) \in E$ for all $i \in N$. Furthermore, for every $i, j \in N$, if $(i, j) \in E$, then $(j, i) \in E$ as well. We denote the set of neighbors of agent $i \in N$ in G by N_i, that is, $N_i = \{j \in N \mid (i, j) \in E\}$. For ease of exposition, we distinguish between the sets N_i and $N_i \setminus \{i\}$, and refer to the latter as the set of *true neighbors* of agent i. Note that for every connected graph $G = \langle N, E\rangle$ with $|N| \geq 2$ and under the assumption that $(i, i) \in E$ for all $i \in N$, it follows that $|N_i| \geq 2$ for all $i \in N$.

An agent i is called a boundary agent if they have exactly one true neighbor, that is, $|N_i \setminus \{i\}| = 1$. We denote the set of boundary agents by B, the unique neighbor of such a boundary agent $i \in B$ by $v(i)$, and the collection of all such neighbors of B by $\nu(B)$, that is, $\nu(B) = \{\nu(i) \mid i \in B\}$.

Each agent $i \in N$ reports a collection of preferences of their neighbors, called a neighbor preference. Formally, a neighbor preference is $\underset{\sim}{P}_i = \left(\underset{\sim}{P}_{i,j}\right)_{j \in N_i}$ such that $\underset{\sim}{P}_{i,j} \in \mathcal{P}$ for all $j \in N_i$. Note that since we have assumed $(i, i) \in E$ for all $i \in N$, it follows that $i \in N_i$. Consequently, each agent i reports their own preference along with the preferences of all their true neighbors in $\underset{\sim}{P}_i$. We denote the set of all possible reports of agent i by $\underset{\sim}{\mathcal{P}}_i$ which is equal to $\mathcal{P}^{|N_i|}$. A neighbor preference profile $\underset{\sim}{P}_N$ is a collection $\left(\underset{\sim}{P}_i\right)_{i \in N}$. We denote the set of all possible neighbor preference profiles by $\underset{\sim}{\mathcal{P}}_N$.

From the set of neighbor preference profiles, we distinguish the set of consistent reports as the profiles where each agent's report about own preferences is completely *backed* by reports from all of their neighbors. A neighbor preference profile $\underset{\sim}{P}_N$ is consistent if for all $i \in N$, $\underset{\sim}{P}_{i,i} = \underset{\sim}{P}_{j,i}$ for all $j \in N_i$. The set of consistent neighbor profiles is denoted by $\widehat{\underset{\sim}{\mathcal{P}}}_N \subsetneq \underset{\sim}{\mathcal{P}}_N$. For a consistent neighbor profile $\widehat{\underset{\sim}{P}}_N \in \widehat{\underset{\sim}{\mathcal{P}}}_N$, we use the notation $\widehat{P}_N \equiv (\widehat{P}_i)_{i \in N}$ to denote the unique preference profile induced by $\widehat{\underset{\sim}{P}}_N$, that is, $\widehat{P}_i = \widehat{\underset{\sim}{P}}_{i,i}$ for all $i \in N$. Thus, a consistent neighbor profile is of the form $\widehat{\underset{\sim}{P}}_N = (\underset{\sim}{P}_i)_{i \in N}$, where, for each agent $i \in N$, $\underset{\sim}{P}_i = (\widehat{P}_j)_{j \in N_i}$.

Given a preference profile $\widehat{P}_N$, we use the notation $\rho_i(\widehat{P}_N)$ to denote the report by agent i about their neighbor according to the profile $\widehat{P}_N$, that is, $\rho_i(\widehat{P}_N) = (\widehat{P}_j)_{j \in N_i}$. Similarly, for any group of agents $M \subseteq N$, we use $\rho_M(\widehat{P}_N)$ to denote the reports by agents $i \in M$ according to the profile about their neighbors, that is $\rho_M(\widehat{P}_N) \equiv \left(\rho_i(\widehat{P}_N)\right)_{i \in M}$

In accordance with the usual literature on social choice, we define a social choice function as a mapping $f : \mathcal{P}^n \to A$ from the set of preference profiles to the set of alternatives. A mechanism $m : \underset{\sim}{\mathcal{P}}_N \to A$ maps an alternative for each profile of neighbor preferences. Note that although the domain of the SCF f is a product set, the domain of the mechanism m is a power set. This is because the domain of neighbor preferences of each agent depends on the number of their neighbors and hence $\underset{\sim}{\mathcal{P}}_i = \underset{\sim}{\mathcal{P}}_j$ if and only if $|N_i| = |N_j|$.

A mechanism $m : \underset{\sim}{\mathcal{P}}_N \to A$ is said to be incentive compatible if, at every consistent neighbor preference profile, no agent can benefit by unilaterally misreporting their neighbor preference.

Definition 4.2.1 *A mechanism $m : \underset{\sim}{\mathcal{P}}_N \to A$ is is said to be **incentive compatible (IC)** if for every $i \in N$, every $P_N \in \mathcal{P}_N$, and every $\underset{\sim}{P}'_i \in \underset{\sim}{\mathcal{P}}_i$, we have $m(\rho_N(P_N)) \; R_i \; m(\underset{\sim}{P}'_i, \rho_{-i}(P_N))$.*

A mechanism $m : \underset{\sim}{\mathcal{P}}_N \to A$ *extends* an SCF $f : \mathcal{P}_N \to A$ if, at every consistent neighbor preference profile, the outcome given by the mechanism m is same as the SCF f at the corresponding induced preference profile. An SCF f is said to be incentive compatible extendable if, there exists an incentive compatible mechanism m that *extends* f.

Definition 4.2.2 *An SCF $f : \mathcal{P}^n \to A$ is said to be* ***incentive compatible extendable (ICE)*** *if there exists an IC mechanism $m : \underset{\sim}{\mathcal{P}}_N \to A$ such that for every $P_N \in \mathcal{P}^n$, we have $f(P_N) = m(\rho_N(P_N))$.*

We present an example of well known rule to illustrate ICE.

Example 4.2.3 *Let $N = \{1, 2, 3, 4\}$ and $A = \{a, b, c, d\}$. Let the network connecting the agents be a line graph with agents 1 and 4 as the boundary agents. That is $(1, 2), (2, 3), (3, 4) \in E$. Consider the following preferences.*

P'_1	P_1	P_2	P_3	P_4
d	b	c	a	b
b	a	a	b	a
c	c	b	c	c
a	d	d	d	d

Table 4.2.1: Preferences of agents

We consider the majority rule with the following tie-breaking order: $a \prec b \prec c \prec d$.

The outcomes are

$$f(P_N) = b, \quad f(P'_1, P_{-1}) = a$$

ICE implies that every IC m must satisfy

$$m(\rho_N(P_N)) = f(P_N) = b \text{ and } m(\rho_N(P'_1, P_{-1})) = f(P'_1, P_{-1}) = a.$$

Agent 2 can deviate from the $\rho_N(P_N)$ to misreport $\underset{\sim}{P}'_2$ where $\underset{\sim}{P}'_{2,1} = P'_1$, $\underset{\sim}{P}'_{2,2} = P_2$, and $\underset{\sim}{P}'_{2,3} = P_3$. IC implies that $m(\rho_N(P_N))\, R_2\, m(\underset{\sim}{P}'_2, \rho_{-2}(P_N))$. That is $b\, R_2\, m(\underset{\sim}{P}'_2, \rho_{-2}(P_N))$. Given P_2, this implies that

$$m(\underset{\sim}{P}'_2, \rho_{-2}(P_N)) = \{b, d\} \tag{4.1}$$

Similarly, agent 1 can deviate from the $\rho_N(P'_1, P_{-1})$ *to misreport* $\underset{\sim}{P}''_1$ *where* $\underset{\sim}{P}''_{1,1} = P_1$ *and* $\underset{\sim}{P}''_{1,2} = P'_2$. *IC implies that* $m(\rho_N(P'_1, P_{-1}))\, R'_1\, m(\underset{\sim}{P}''_1, \rho_{-1}(P'_1, P_{-1}))$. *That is a* $R'_1\, m(\underset{\sim}{P}''_1, \rho_{-1}(P'_1, P_{-1}))$. *Given* P'_1, *this implies that*

$$m(\underset{\sim}{P}''_1, \rho_{-1}(P'_1, P_{-1})) = \{a\} \tag{4.2}$$

Note that the way the two profiles, $(\underset{\sim}{P}'_2, \rho_{-2}(P_N))$ *and* $(\underset{\sim}{P}''_1, \rho_{-1}(P'_1, P_{-1}))$*are constructed, they are identical. This combined with Equations 4.1, 4.2 imply that* $m(\underset{\sim}{P}'_2, \rho_{-2}(P_N)) = \emptyset$. *Therefore, majority rule fails to satisfy ICE.*

4.3 Results

We analyze the structure of ICE SCFs based on the properties of the underlying graph $G = \langle N, E\rangle$. Our results are divided into three distinct subsections. The first subsection considers graphs where every agent $i \in N$ has at least three neighbors. The second focuses on the case where the graph consists of exactly two agents connected to each other. The final subsection addresses arbitrary connected graphs with more than two agents in which at least one agent has fewer than three neighbors. These three cases are mutually disjoint and together exhaust all possible structures of connected graphs.

4.3.1 Graph with minimum degree at least three

The following theorem is notably strong: it asserts that any social choice function can be extended in an incentive-compatible manner, provided that each agent has at least three neighbors — that is, when there is sufficient public knowledge about agents' preferences, making it more difficult for an individual to manipulate the outcome.

Theorem 4.3.1 *Let the network G be such that* $|N_i| \geq 3$ *for all* $i \in N$. *Then, every social choice function* $f : \mathcal{P}^n \to A$ *is ICE.*

Proof: Consider an arbitrary SCF $f : \mathcal{P}^n \to A$. We construct a mechanism $m : \underset{\sim}{\mathcal{P}}_N \to A$ such that m is IC and m extends f. Consider the mechanism m such that $m(\rho_N(P_N)) = m(\underset{\sim}{P}'_i, \rho_{-i}(P_N))$ for all $i \in N$, for all $P_N \in \mathcal{P}_N$, and for all $\underset{\sim}{P}'_i \in \underset{\sim}{\mathcal{P}}_i$, and $m(\rho_N(P_N)) = f(P_N)$. By definition, m is IC and m extends f which completes the proof.

It remains to show that m is well defined. It is sufficient to show that there do not exist two distinct consistent neighbor profiles such that unilateral deviations by two (possibly different) agents from each profile yield an identical resulting neighbor profile. Suppose for contradiction, we assume that there exist two distinct $P_N, \bar{P}_N \in \mathcal{P}_N$ such that for some $i, j \in N$, we have

$$\underset{\sim}{P}'_N \equiv (\underset{\sim}{P}'_i, \rho_{-i}(P_N)) = (\underset{\sim}{P}''_j, \rho_{-j}(\bar{P}_N)) \equiv \underset{\sim}{P}''_N \tag{4.3}$$

for some $\underset{\sim}{P}'_i \in \underset{\sim}{\mathcal{P}}_i$ and $\underset{\sim}{P}''_j \in \underset{\sim}{\mathcal{P}}_j$.

Since $P_N \neq \bar{P}_N$, there exists an agent $k \in N$ such that $P_k \neq \bar{P}_k$. If $k \in N \setminus \{i, j\}$, then the report of agent k in $\underset{\sim}{P}'_N \equiv (\underset{\sim}{P}'_i, \rho_{-i}(P_N))$ is $\rho_k(P_N)$, that is $\underset{\sim}{P}'_k = \rho_k(P_N)$. This implies that $\underset{\sim}{P}'_{k,k} = P_k$. Similarly the report of agent k in $\underset{\sim}{P}''_N \equiv (\underset{\sim}{P}''_j, \rho_{-j}(\bar{P}_N))$ is $\rho_k(\bar{P}_N)$, that is $\underset{\sim}{P}'_k = \rho_k(\bar{P}_N)$. This implies that $\underset{\sim}{P}''_{k,k} = \bar{P}_k$. Since $P_k \neq \bar{P}_k$, we have $\underset{\sim}{P}'_{k,k} \neq \underset{\sim}{P}''_{k,k}$ which implies $\underset{\sim}{P}'_k \neq \underset{\sim}{P}''_k$ which contradicts the equality of Equation 4.3. Hence k must belong to $\{i, j\}$. Without loss of generality, assume that $k = i$ which implies that $P_i \neq \bar{P}_i$. Since $|N_i| \geq 3$, there exists an agent $l \in N_i$ distinct from both i and j (if $j \in N_i$). The report of agent l in the neighbor profile $\underset{\sim}{P}'_N$ is $\rho_l(P_N)$ which implies that $\underset{\sim}{P}'_{l,i} = P_i$.

Similarly, the report of agent l in the neighbor profile $\underset{\sim}{P}''_N$ is given by $\rho_l(\bar{P}_N)$, which implies that $\underset{\sim}{P}''_{l,i} = \bar{P}_i$. Since $P_i \neq \bar{P}_i$, it follows that $\underset{\sim}{P}'_l \neq \underset{\sim}{P}''_l$, contradicting Equation 4.3. An analogous argument applies in the case where $k = j$. This completes the proof of the theorem.

■

4.3.2 The case of two agents

We introduce the notion of the lower contour set intersection property. This property requires that, for any two preference profiles, there exists at least one common alternative that is strictly worse for agent 1 than the outcome under the first profile and strictly worse for agent 2 than the outcome under the second profile. Intuitively, such an alternative serves as a threat to agent 1 in the first profile and to agent 2 in the second, thereby playing a role in limiting incentives to misreport preferences.

Definition 4.3.2 *A social choice funtion $f : \mathcal{P}^2 \to A$ is said to satisfy the* ***lower contour set intersection***

property (LCSIP) *if for every pair of preference profiles* (P_1, P_2) *and* $(P'_1, P'_2) \in \mathcal{P}^2$*, we have*

$$L(f(P_1, P_2), P_1) \cap L(f(P'_1, P'_2), P'_2) \neq \emptyset \tag{4.4}$$

Our next theorem establishes that, in the case of two agents, the lower contour set intersection property (LCSIP) is both a necessary and sufficient condition for a social choice function to be extended in an incentive-compatible manner.

Theorem 4.3.3 *A social choice function $f : \mathcal{P}^2 \to A$ is incentive compatible extendable if and only if it satisfies the LCSIP.*

The proof of this theorem is similar to the proof of Theorem 4.3.8.

For any $P \in \mathcal{P}$, P^c denotes the preference with complete reverse ordering. Thus aPb if and only if $bP^c a$ for all $a, b \in A$. We define the set $OP \subsetneq \mathcal{P}^2$ as the set of all preference profiles such that the preference of one agent is the complete reverse of the other agent's preference, that is,
$OP = \{(Q_1, Q_2) \in \mathcal{P}^2 \mid Q_2 = Q_1^c\}$.

Note that LCSIP requires verifying that Equation 4.4 holds for every pair of preference profiles, which can be computationally demanding. The following proposition simplifies this task by providing a much weaker condition that still implies LCSIP. Specifically, it identifies a smaller set of preference profiles based on opposite preferences such that the validity of Equation 4.4 for these pairs guarantees LCSIP.

Proposition 4.3.1 *Suppose that for every $(P_1, P_2) \in \mathcal{P}^2$ and every $(\bar{P}_1, \bar{P}_2) \in OP$, we have*

$$L(f(P_1, P_2), P_1) \cap L(f(\bar{P}_1, \bar{P}_2), \bar{P}_2) \neq \emptyset \quad \textit{and} \quad L(f(P_1, P_2), P_2) \cap L(f(\bar{P}_1, \bar{P}_2), \bar{P}_1) \neq \emptyset.$$

Then the social choice function $f : \mathcal{P}^2 \to A$ satisfies LCSIP.

Proof: We prove the proposition through contradiction. Assume that for every $(P_1, P_2) \in \mathcal{P}^2$, and every $(\bar{P}_1, \bar{P}_2) \in OP$,

$$L(f(P_1, P_2), P_1) \cap L(f(\bar{P}_1, \bar{P}_2), \bar{P}_2) \neq \emptyset \quad \text{and} \quad L(f(P_1, P_2), P_2) \cap L(f(\bar{P}_1, \bar{P}_2), \bar{P}_1) \neq \emptyset.$$

holds true, but $f: \mathcal{P}^2 \to A$ does not satisfy LCSIP. This implies that there exists a pair of preference profiles (P_1, P_2) and $(P'_1, P'_2) \in \mathcal{P}^2$ such that

$$L(f(P_1, P_2), P_1) \cap L(f(P'_1, P'_2), P'_2) = \emptyset. \tag{4.5}$$

Consider the $(\bar{P}_1, \bar{P}_2) \in OP$ such that $L(f(P_1, P_2), P_1) = L(f(P_1, P_2), \bar{P}_1)$. Since $(P_1, P_2), (P'_1, P'_2) \in \mathcal{P}^2$ and $(\bar{P}_1, \bar{P}_2) \in OP$, by our assumption we must have

$$L(f(P_1, P_2), P_1) \cap L(f(\bar{P}_1, \bar{P}_2), \bar{P}_2) \neq \emptyset. \tag{4.6}$$

and

$$L(f(P'_1, P'_2), P'_2) \cap L(f(\bar{P}_1, \bar{P}_2), \bar{P}_1) \neq \emptyset. \tag{4.7}$$

Since $(\bar{P}_1, \bar{P}_2) \in OP$, we have $\bar{P}_2 = \bar{P}_1^c$. This implies that for any alternative $a \in A$, we have $L(a, \bar{P}_2) = U(a, \bar{P}_1)$. Therefore $L(f(\bar{P}_1, \bar{P}_2), \bar{P}_2) = U(f(\bar{P}_1, \bar{P}_2), \bar{P}_1)$. But we know from the construction of $\bar{P}_1$, $L(f(P_1, P_2), P_1) = L(f(P_1, P_2), \bar{P}_1)$. Therefore Equation 4.6 reduces to $L(f(P_1, P_2), \bar{P}_1) \cap U(f(\bar{P}_1, \bar{P}_2), \bar{P}_1) \neq \emptyset$. This can only be possible when $f(P_1, P_2)\bar{R}_1 f(\bar{P}_1, \bar{P}_2)$. This implies that $L(f(\bar{P}_1, \bar{P}_2), \bar{P}_1) \subseteq L(f(P_1, P_2), \bar{P}_1)$. But we already know that $L(f(P_1, P_2), P_1) = L(f(P_1, P_2), \bar{P}_1)$ which implies that $L(f(\bar{P}_1, \bar{P}_2), \bar{P}_1) \subseteq L(f(P_1, P_2), P_1)$. Combining this with Equation 4.7, we have $L(f(P'_1, P'_2), P'_2) \cap L(f(P_1, P_2), \bar{P}_1) \neq \emptyset$ which contradicts Equation 4.5. This completes the proof of the proposition.

■

4.3.3 Arbitrary Connected Graph

In this section, we consider arbitrary connected graphs over an arbitrary number of agents. We generalize the lower contour set intersection property (LCSIP) to such graph structures and provide a characterization of social choice functions that are extendable in an incentive-compatible manner, based on this generalized version of the LCSIP.

Consider an arbitrary graph $G = \langle N, E \rangle$ with no isolated nodes and with $|N| > 2$. It is worth noting that, in the case of two agents, each agent has exactly one true neighbor—namely, the other. More generally, as indicated by Theorem 4.3.1, the challenge of achieving an incentive-compatible extension arises primarily for agents with exactly one true neighbor. This is because their preferences are not sufficiently known by others to be effectively verified. The next concept, referred to as the generalized intersection condition, intuitively ensures that the LCSIP holds even for such agents by requiring the existence of a common alternative that can act as a threat to both the agent and their sole neighbor.

Definition 4.3.4 *An SCF $f : \mathcal{P}^n \to A$ satisfies **generalized intersection condition** (GIC) if for all $i \in B$ and for all $P_i, P_i' \in \mathcal{P}$, and $P_{-i} \in \mathcal{P}^{n-1}$, we have $L(f(P_i, P_{-i}), P_i) \cap L(f(P_i', P_{-i}), P_{\nu(i)}) \neq \emptyset$.*

We present a few examples to illustrate generalized intersection condition.

Example 4.3.5 *Let $N = \{1, 2, 3, 4, 5\}$ and $A = \{a, b, c, d\}$. Let the network connecting the agents be a line graph with agents 1 and 5 as the boundary agents. That is $(1, 2), (2, 3), (3, 4), (4, 5) \in E$. Consider the following preferences.*

P_1'	P_1	P_2	P_3	P_4	P_5
d	b	c	d	b	a
b	a	d	a	d	b
a	c	b	c	c	c
c	d	a	b	a	d

Table 4.3.1: Preferences of agents

We consider the anti majority rule with tie-breaking order: $a \prec b \prec c \prec d$. The anti majority rule is the Borda rule where the score for the last ranked alternative in a preference is -1 and the scores for all other ranked alternatives in a preference is 0. In other words, at any preference profile, the cumulative score of each alternative is obtained by summing its scores across all agents' preferences. The outcome is then chosen as the alternative with the highest cumulative score. The outcomes are

$$f(P_N) = c, \quad f(P_1', P_{-1}) = b$$

Generalized intersection condition fails since

$$L(f(P_1, P_{-1}), P_1) = \{c, d\}, \quad L(f(P_1', P_{-1}), P_2) = \{b, a\}.$$

This implies that the two sets are disjoint.

Example 4.3.6 *Let $N = \{1, 2, 3, 4\}$ and $A = \{a, b, c, d\}$. Let the network connecting the agents be a line graph with agents 1 and 4 as the boundary agents. Let the social choice function be defined as follows:*

$$f(P_N) = \begin{cases} r_1(P_2), & \text{if } r_1(P_2) = r_1(P_3), \\ r_1(P_4), & \text{otherwise}. \end{cases}$$

We show that this social choice function satisfies the generalized intersection condition (GIC). To this end, it suffices to verify that f satisfies the condition in Definition 4.3.4 for the boundary agents, namely agents 1 and 4.

Agent 1. *Observe first that f is independent of the preference of agent 1. Hence, for any P_{-1}, we have*

$$f(P_1, P_{-1}) = f(P_1', P_{-1}) \quad \text{for all } P_1, P_1'.$$

By the definition of the lower contour set, it follows that

$$f(P_1, P_{-1}) \in L(f(P_1, P_{-1}), P_1) \quad \text{and} \quad f(P_1', P_{-1}) \in L(f(P_1', P_{-1}), P_{\nu(1)}).$$

Since $f(P_1, P_{-1}) = f(P_1', P_{-1})$, we obtain

$$f(P_1, P_{-1}) \in L(f(P_1, P_{-1}), P_1) \cap L(f(P_1', P_{-1}), P_{\nu(1)}),$$

which implies that

$$L(f(P_1, P_{-1}), P_1) \cap L(f(P_1', P_{-1}), P_{\nu(1)}) \neq \emptyset.$$

Thus, GIC is satisfied for agent 1.

Agent 4. *Consider first a preference profile P_N such that $r_1(P_2) = r_1(P_3)$. In this case, f is independent of P_4, that* *is,*

$$f(P_4, P_{-4}) = f(P'_4, P_{-4}) \quad \textit{for all } P_4, P'_4.$$

Using the same argument as for agent 1, it follows that GIC is satisfied in this case.

Now consider a preference profile P_N such that $r_1(P_2) \neq r_1(P_3)$. Then $f(P_N) = r_1(P_4)$. Hence,

$$L(f(P_4, P_{-4}), P_4) = A,$$

which implies that

$$L(f(P_4, P_{-4}), P_4) \cap L(f(P'_4, P_{-4}), P_{\nu(4)}) \neq \emptyset.$$

Therefore, GIC is satisfied for agent 4.

Example 4.3.7 *We now focus on a special class of graphs—the star graph—where a central node is connected to all others, and peripheral nodes are not connected to each other. Formally, $G = \langle N, E \rangle$ with:*

$$E = \{(1, j) : j \in N \setminus \{1\}\}$$

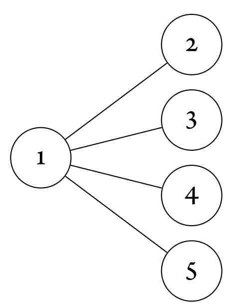

Figure 4.3.1: A star graph with agent 1 at the center connected to agents 2–5

Under this network structure, the set of boundary agents B is equal to $N \setminus \{1\}$. Thus the generalized intersection condition reduces to

$$L(f(P_i, P_{-i}), P_i) \cap L(f(P'_i, P_{-i}), P_1) \neq \emptyset \tag{4.8}$$

for all $i \in N \setminus \{1\}$, for all $P_i, P'_i \in \mathcal{P}$ and for all $P_{-i} \in \mathcal{P}^{n-1}$

Define the social choice function f in the following manner. Let $f(P_N) = x$, where x is such that

$$xP_1y \quad \text{for all } y \in \{r_1(P_j) \mid j \neq 1\}$$

That is, f selects the best alternative for agent 1 among the top-ranked alternatives of all others. We show that f satisfies the generalized intersection condition.

Consider any $i \in N \setminus \{1\}$ and any preference profile P_N and P'_i. If $f(P_N) = f(P'_i, P_{-i})$, then Equation 4.8 is satisfied since $f(P_N) \in L(f(P_i, P_{-i}), P_i)$ and $f(P'_i, P_{-i}) \in L(f(P'_i, P_{-i}), P_1)$.

Suppose that $f(P_N) \neq f(P'_i, P_{-i})$. This implies that either $f(P_N) = r_1(P_i)$ or $f(P'_i, P_{-i}) = r_1(P'_i)$ or both. If $f(P_N) = r_1(P_i)$, then $L(f(P_i, P_{-i}), P_i) = A$ which implies that Equation 4.8 holds true. Else if $f(P'_i, P_{-i}) = r_1(P'_i)$ and $f(P_N) \neq r_1(P_i)$, then $f(P'_i, P_{-i})\, P_1\, f(P_N)$. This implies that $f(P_N) \in L(f(P'_i, P_{-i}), P_1)$. Therefore Equation 4.8 is satisfied. This completes all the cases and hence f satisfies generalized intersection condition.

We present the main theorem of this section. The following theorem characterizes the class of social choice functions that can be extended in an incentive-compatible manner.

Theorem 4.3.8 *An SCF $f : \mathcal{P}^n \to A$ is incentive compatible extendable if and only if it satisfies generalized intersection condition.*

Proof: Only if part : Suppose that the SCF f is incentive compatible extendable. We show that f satisfies generalized intersection condition. Consider arbitrary $i \in B$, $P'_i, P''_i \in \mathcal{P}$, and $P'_{-i} \in \mathcal{P}^{n-1}$. We show that $L(f(P'_i, P'_{-i}), P'_i) \cap L(f(P''_i, P'_{-i}), P'_{\nu(i)}) \neq \emptyset$. Since f is incentive compatible extendable, there exists an incentive compatible mechanism $m : \underset{\sim}{\mathcal{P}}_N \to A$ such that for any preference profile P_N, we have $f(P_N) = m(\rho_N(P_N))$. Consider any arbitrary agent $i \in B$, any two arbitrary consistent neighbor profiles $\rho_N(P'_N)$ and $\rho_N(P''_i, P'_{-i})$. Since m extends f, we have $m(\rho_N(P'_N)) = f(P'_N)$ and $m(\rho_N(P''_i, P'_{-i})) = f(P''_i, P'_{-i})$. Further, since m is incentive compatible, considering the misreport of i from $\rho_i(P'_N)$ to $\underset{\sim}{P}_i$, we have

$$m\big(\rho_i(P'_N), \rho_{-i}(P'_N)\big)\; R'_i\; m\big(\underset{\sim}{P}_i, \rho_{-i}(P'_N)\big). \quad \text{for all } \underset{\sim}{P}_i \in \underset{\sim}{\mathcal{P}}_i \tag{4.9}$$

Similarly, considering the misreport of agent $\nu(i)$ from $\rho_{\nu(i)}(P_i^{''}, P_{-i}^{'})$ to $\underset{\sim}{P}_{\nu(i)}$, we obtain

$$m\big(\rho_{\nu(i)}(P_i^{''}, P_{-i}^{'}), \rho_{-\nu(i)}(P_i^{''}, P_{-i}^{'})\big)\; R_{\nu(i)}^{'}\; m\big(\underset{\sim}{P}_{\nu(i)}, \rho_{-\nu(i)}(P_i^{''}, P_{-i}^{'})\big). \quad \text{for all } \underset{\sim}{P}_{\nu(i)} \in \underset{\sim}{\mathcal{P}}_{\nu(i)} \tag{4.10}$$

Now consider the report $\bar{\underset{\sim}{P}}_i$ of agent i according to the profile $(P_i^{''}, P_{-i}^{'})$, that is, $\bar{\underset{\sim}{P}}_i = \rho_i(P_i^{''}, P_{-i}^{'})$ and the report of the unique neighbor $\bar{\underset{\sim}{P}}_{\nu(i)}$ of agent i at the profile $(P_N^{'})$, that is, $\bar{\underset{\sim}{P}}_{\nu(i)} = \rho_{\nu(i)}(P_N^{'})$. Putting $\bar{\underset{\sim}{P}}_i$ in Equation 4.9, and using the fact that $m(\rho_N(P_N^{'})) = f(P_N^{'})$, we have $f(P_N^{'})\; R_i^{'}\; m(\bar{\underset{\sim}{P}}_i, \rho_{-i}(P_N^{'}))$, which implies

$$m(\bar{\underset{\sim}{P}}_i, \rho_{-i}(P_N^{'})) \in L(f(P_N^{'}), P_i^{'}). \tag{4.11}$$

Similarly, putting $\bar{\underset{\sim}{P}}_{\nu(i)}$ in Equation 4.10, and using the fact that $m(\rho_N(P_i^{''}, P_{-i}^{'})) = f(P_i^{''}, P_{-i}^{'})$, we have $f(P_i^{''}, P_{-i}^{'})\; R_{\nu(i)}^{'}\; m(\bar{\underset{\sim}{P}}_{\nu(i)}, \rho_{-\nu(i)}(P_i^{''}, P_{-i}^{'}))$, which implies

$$m(\bar{\underset{\sim}{P}}_{\nu(i)}, \rho_{-\nu(i)}(P_i^{''}, P_{-i}^{'})) \in L(f(P_i^{''}, P_{-i}^{'}), P_{\nu(i)}^{'}). \tag{4.12}$$

Observe that in the neighbor profiles $(\bar{\underset{\sim}{P}}_i, \rho_{-i}(P_N^{'}))$ and $(\bar{\underset{\sim}{P}}_{\nu(i)}, \rho_{-\nu(i)}(P_i^{''}, P_{-i}^{'}))$ all agents in N except i and their neighbor $\nu(i)$ report the same neighbor preferences about their neighbor across the two profiles, that is, $\rho_j(P_N') = \rho_j(P_i^{''}, P_{-i}^{'})$ for all $j \in N \setminus \{i, \nu(i)\}$. This is because, only the preference of agent i has been changed from $P_i^{'}$ in $\rho_j(P_N')$ to $P_i^{''}$ in $\rho_j(P_i^{''}, P_{-i}^{'})$, and since i has exactly one neighbor $\nu(i)$, this will not affect the reports of the agents other than i and $\nu(i)$ across the two neighbor profiles $(\bar{\underset{\sim}{P}}_i, \rho_{-i}(P_N^{'}))$ and $(\bar{\underset{\sim}{P}}_{\nu(i)}, \rho_{-\nu(i)}(P_i^{''}, P_{-i}^{'}))$. Since $\bar{\underset{\sim}{P}}_i = \rho_i(P_i^{''}, P_{-i}^{'})$, the report of agent i at the neighbor preference profile $(\bar{\underset{\sim}{P}}_i, \rho_{-i}(P_N^{'}))$ about the preference of themselves and their neighbor $\nu(i)$ are $P_i^{''}$ and P'_{-i}, respectively. Furthermore, from the structure of the neighbor profile $(\bar{\underset{\sim}{P}}_{\nu(i)}, \rho_{-\nu(i)}(P_i^{''}, P_{-i}^{'}))$, it follows straight forwardly that the report of agent i about the preference of themselves and neighbor $\nu(i)$ are $P_i^{''}$ and P_i', respectively. From the definition of $\bar{\underset{\sim}{P}}_{\nu(i)}$, it follows that the report of the agent $\nu(i)$ about any agent $j \neq i$ will be the same in the two neighbor profiles $(\bar{\underset{\sim}{P}}_i, \rho_{-i}(P_N^{'}))$ and $(\bar{\underset{\sim}{P}}_{\nu(i)}, \rho_{-\nu(i)}(P_i^{''}, P_{-i}^{'}))$. Furthermore, since $\bar{\underset{\sim}{P}}_{\nu(i)} = \rho_{\nu(i)}(P_N')$, the report of agent $\nu(i)$ about the preference of agent i is also the same in the neighbor profiles $(\bar{\underset{\sim}{P}}_i, \rho_{-i}(P_N^{'}))$ and $(\bar{\underset{\sim}{P}}_{\nu(i)}, \rho_{-\nu(i)}(P_i^{''}, P_{-i}^{'}))$. Thus, it follows that the full report of the agent $\nu(i)$ remains the same across the two neighbor preference profiles $(\bar{\underset{\sim}{P}}_i, \rho_{-i}(P_N^{'}))$ and $(\bar{\underset{\sim}{P}}_{\nu(i)}, \rho_{-\nu(i)}(P_i^{''}, P_{-i}^{'}))$.

All these together imply that the reports of all agents remain the same in the two neighbor profiles $(\underset{\sim}{\bar{P}}_i, \rho_{-i}(P'_N))$ and $(\underset{\sim}{\bar{P}}_{\nu(i)}, \rho_{-\nu(i)}(P''_i, P'_{-i}))$, which implies that they are the same network profiles, that is, $(\underset{\sim}{\bar{P}}_i, \rho_{-i}(P'_N)) = (\underset{\sim}{\bar{P}}_{\nu(i)}, \rho_{-\nu(i)}(P''_i, P'_{-i}))$. Therefore, $m(\underset{\sim}{\bar{P}}_i, \rho_{-i}(P'_N)) = m(\underset{\sim}{\bar{P}}_{\nu(i)}, \rho_{-\nu(i)}(P''_i, P'_{-i}))$. Using this in Equations 4.11 and 4.12, we obtain
$m(\underset{\sim}{\bar{P}}_{\nu(i)}, \rho_{-\nu(i)}(P''_i, P'_{-i})) \in L(f(P'_i, P'_{-i}), P'_i) \cap L(f(P''_i, P'_{-i}), P'_{\nu(i)})$. This completes the only if part of the proof.

If part : Suppose that the SCF $f : \mathcal{P}^n \to A$ satisfies generalized intersection condition. We show that f is ICE. Since f satisfies generalized intersection condition, for all $i \in B$ and for all $P_i, P'_i \in \mathcal{P}$, and $P_{-i} \in \mathcal{P}^{n-1}$, we have $L(f(P_i, P_{-i}), P_i) \cap L(f(P'_i, P_{-i}), P_{\nu(i)}) \neq \emptyset$. Suppose $x \in L(f(P_i, P_{-i}), P_i) \cap L(f(P'_i, P_{-i}), P_{\nu(i)})$. We will now construct a mechanism $m : \underset{\sim}{\mathcal{P}}_N \to A$ such that f is ICE through m.

Recall that B is the set of agents who have exactly one true neighbor, and $\nu(B)$ is the set of unique neighbors $\nu(i)$ of the agents i in B; that is, $\nu(B) = \{\nu(i) \mid i \in B\}$. At every consistent neighbor preference profile, say $\rho_N(P_N)$, let us define $m(\rho_N(P_N)) = f(P_N)$. At every (inconsistent) neighbor preference profile $(\underset{\sim}{P}'_i, \rho_{-i}(P_N))$ where only an agent $i \in N \setminus B \cup \nu(B)$ unilaterally deviates from a consistent neighbor profile $(\rho_N(P_N))$, define $m(\underset{\sim}{P}'_i, \rho_{-i}(P_N)) = m(\rho_N(P_N))$. We now consider a neighbor preference profile $(\underset{\sim}{P}'_i, \rho_{-i}(P_N))$ where an agent $i \in B \cup \nu(B)$ deviates from a consistent neighbor preference profile $(\rho_N(P_N))$ to a preference $\underset{\sim}{P}'_i$. If $\underset{\sim}{P}'_i$ and $\rho_{\nu(i)}(P_N)$ do not differ in terms of their report of the preference of the agent in B, then define $m(\underset{\sim}{P}'_i, \rho_{-i}(P_N)) = m(\rho_N(P_N))$. Otherwise, define $m(\underset{\sim}{P}'_i, \rho_{-i}(P_N))$ as an arbitrary element in the set $L(f(P_N), P_i) \cap L(f(P'_i, P_{-i}), P_{\nu(i)})$ which is non empty by the generalized intersection condition.

The fact that the mechanism m defined above is well defined follows in the same manner as we have shown the mechanism m defined in Theorem 4.3.1 is well defined. Since $m(\rho_N(P_N)) = f(P_N)$ for every P_N, the mechanism m extends f. We proceed to show that the mechanism m is IC.

For any agent $i \in N \setminus B \cup \nu(B)$, $m(\underset{\sim}{P}'_i, \rho_{-i}(P_N)) = m(\rho_N(P_N))$ which implies that agent i cannot manipulate m. Consider an agent $i \in B \cup \nu(B)$ and a unilateral deviation of i from a network consistent profile $\rho_N(P_N)$ to a preference $\underset{\sim}{P}'_i$. If $\underset{\sim}{P}'_i$ and $\rho_{\nu(i)}(P_N)$ do not differ in terms of their report of the preference of the agent in B, then we have $m(\underset{\sim}{P}'_i, \rho_{-i}(P_N)) = m(\rho_N(P_N))$, which implies that agent i cannot manipulate m. Finally consider the case where $\underset{\sim}{P}'_i$ and $\rho_{v(i)}(P_N)$ differ in terms of their report of the preference of the agent in B. From the construction of m, we have $m(\rho_N(P_N)) = f(P_N)$ and $m(\underset{\sim}{P}'_i, \rho_{-i}(P_N)) \in L(f(P_i, P_{-i}), P_i) \cap L(f(P'_i, P_{-i}), P_{v(i)})$. In particular, $m(\underset{\sim}{P}'_i, \rho_{-i}(P_N)) \in L(f(P_i, P_{-i}), P_i)$, which implies $f(P_i, P_{-i})\ R_i\ m(\underset{\sim}{P}'_i, \rho_{-i}(P_N))$ and hence $m(\rho_N(P_N)) R_i m(\underset{\sim}{P}'_i, \rho_{-i}(P_N))$. This proves that m is IC and hence the proof is complete.

■

5

Matching under multiple single-peaked domains

5.1 Introduction

5.1.1 Description and background of problem

We look at the housing problem introduced in [28]. There are n houses and n agents for some integer $n \geq 2$. Initially, each agent owns a unique house and hence different agents own different houses. Agents have strict preferences over the houses. A matching rule reallocates the houses amongst the agents based on the preferences of the agents over the houses. A social designer's problem is to find matching rules that satisfy certain desirable properties. The matching literature ([29], [27], [15]) focuses predominantly on three desirable properties for such matching rules: Individual rationality (IR), Pareto optimality (PO), and strategy-proofness(SP). IR requires that the house allocated to an agent is at least as good as their initial endowed house. PO guarantees that there is no "better" way to allocate the houses, that is, there is no other way to allocate the houses so that every agent will be weakly better off and some agents will be strictly better off. Strategy-proofness ensures that no agent can (unilaterally) misreport their preference and get a strictly better allocation.

When agents' preferences over the set of houses are allowed to be unrestricted, [24], [23], and [15] show that Gale's Top Trading Cycle (TTC) algorithm is the only matching rule that satisfies these three conditions. However the uniqueness of the result does not hold when the domain of preferences is restricted. [2] provides a different algorithm called the Crawler, which satisfies IR, PO, and strategy-proofness on the domain of single-peaked preferences. This has further inspired studies on other restricted preference domains to explore the nature of matching rules that satisfy these three conditions (e.g. [30]).

In this paper, we consider the multiple single-peaked domain introduced in [22]. Single-peaked preferences are defined with respect to an underlying ordering that is common to all agents, such as house size or distance from the nearest public school. A multiple single-peaked domain is the union of several single-peaked domains corresponding to different underlying orderings. This framework captures settings in which there is no single objective way to order the houses. Instead, each agent may have a distinct subjective perception of the ordering; nevertheless, given their individual ordering, preferences remain single-peaked.

Multiple single-peaked preferences capture a wide variety of preference domains. If all the agents' orderings are same, then we have the usual single-peaked domain. On the other extreme, if we allow for every possible agents' orderings, then we have the unrestricted domain. However in reality, agents' personal understanding of the orderings are often not too dissimilar from each other. Agents agree on the basic structure but differ in the exact positioning of some of the houses in their underlying orderings. More precisely, we assume that agents can partition the set of houses into H_L, H_M, H_R such that they agree on the exact positioning of the houses in H_L and H_R and the relative positioning of the houses in H_M with respect to H_L and H_R. However they disagree about the exact positioning of the houses in H_M.

5.1.2 Contribution of the paper

We begin by observing that the standard *Crawler algorithm* relies fundamentally on the existence of a unique linear ordering of alternatives to determine the direction of *crawling*. As such, it is not well-defined in our setting when the set H_M, which consists of houses lacking a common underlying order, is non-empty. The lack of a consistent ordering over H_M breaks the essential assumption needed for the Crawler to operate coherently, thereby limiting its applicability to structured domains.

To overcome this limitation, we introduce a novel matching rule, which we call the *C-TTC algorithm*. This rule synthesizes the structural advantages of the Crawler with the flexibility and generality of the classic *Top Trading Cycles* (TTC) algorithm. Roughly speaking, the C-TTC algorithm applies the Crawler logic to the subsets H_L and H_R—the sets of houses over which a known linear ordering is assumed—and applies the TTC rule to the remaining subset H_M, for which no such ordering exists.

The operation of the C-TTC algorithm proceeds in a stage-wise manner and depends critically on the actions of certain *boundary agents*. These boundary agents are defined as the rightmost agent associated with H_L and the leftmost agent associated with H_R, respectively. In each stage of the algorithm, these boundary agents *point* to their most-preferred available house. The direction in which they point determines the partitioning of the remaining houses: one portion is selected for processing via the TTC component, and the complement is resolved using the Crawler logic. This mechanism ensures that the algorithm dynamically balances between structured and unstructured subdomains during the allocation process.

When $H_M = \emptyset$, the entire house set is ordered, corresponding to a single-peaked domain; in this case, the C-TTC algorithm reduces exactly to the Crawler algorithm. Conversely, when $H_M = H$ (and $H_L = H_R = \emptyset$), the domain is completely unrestricted, and the C-TTC algorithm reduces to the classical TTC algorithm.

We prove that the C-TTC algorithm satisfies three fundamental axiomatic properties across all configurations of H_M: *individual rationality* (IR), *Pareto optimality* (PO), and *strategy-proofness* (SP). In particular, when $H_M = H$, our result recovers the well-known characterization of the TTC algorithm as the unique matching mechanism satisfying IR, PO, and SP in unrestricted domains [15, 23]. However, as demonstrated in Example 5.3.3, when H_M is a strict subset of H, the C-TTC algorithm produces outcomes that differ from those generated by TTC, thereby revealing the impact of the partial ordering embedded in H_L and H_R.

This distinction is of significant theoretical interest. It shows that the C-TTC algorithm is not simply a technical interpolation between the Crawler and TTC rules but represents a genuine generalization. It introduces a new family of mechanisms that satisfy IR, PO, and SP across a broad class of preference domains, including and extending beyond the standard single-peaked structure. As such, our work contributes to a deeper structural understanding of matching mechanisms, revealing how strong incentive and efficiency guarantees can be preserved along a spectrum of domains ranging from fully ordered (single-peaked) to entirely unrestricted.

Obvious Strategy-Proofness and Its Implications

[13] introduced a refinement of the classical notion of strategy-proofness, termed *obvious strategy-proofness* (OSP), and argued that it provides a more behaviorally plausible model of strategic interaction, particularly for agents with limited cognitive abilities. In contrast to strategy-proofness, which is defined purely in terms of the incentive properties of a direct revelation mechanism, obvious strategy-proofness depends crucially on the *extensive-form implementation* of a mechanism.

A matching rule is said to be *obviously strategy-proof* if there exists an extensive-form mechanism implementing the rule, together with a profile of *obviously dominant strategies*. To determine whether a strategy is obviously dominant for an agent i, consider any node in the game tree at which i is called to act according to the strategy. At each such node, agent i evaluates the *worst possible outcome* over all terminal nodes consistent with continuing along the prescribed strategy, given all possible future behaviors of the other agents. This outcome is then compared to the *best possible outcome* that i could achieve by deviating from the strategy at the current node (again, considering all possible future behaviors of others). If, at every such decision point, the worst outcome from following the strategy is at least as good as the best outcome from any deviation, then the strategy is said to be *obviously dominant*.

Theorem 5.4.4 of this paper establishes an impossibility result: there exists *no* matching rule on any non-trivial *multiple single-peaked domain* that simultaneously satisfies *individual rationality* (IR), *Pareto optimality* (PO), and *obvious strategy-proofness*.

This result is particularly striking when contrasted with the known characterizations in simpler domains. On the one hand, the trivial instance of the multiple single-peaked domain—namely, the standard *single-peaked domain*—admits a rich class of matching rules that satisfy IR, PO, and OSP. In particular, [30] provides a complete characterization of all such rules on single-peaked domains. On the other hand, our result shows that as soon as the domain becomes non-trivially multiple-peaked, the existence of any IR, PO, and OSP rule is precluded. This sharp contrast underscores the fragility of obvious strategy-proofness under seemingly mild relaxations of domain structure.

The rest of the paper is organized as follows. Section 5.2 introduces the basic model. In section 5.3, we define and explain the C-TTC algorithm. We then state our first theorem that establishes that the C-TTC algorithm satisfies individual rationality, pareto optimality and strategy-proofness. In section 5.4, we define the notion of obvious strategy-proofness and state our impossibility result regarding the existence of matching rules satisfying IR, PO and obvious strategy-proofness. All proofs are collected in the appendix.

5.2 Preliminaries

5.2.1 Domain of preferences and their properties

Let $N = \{1, \ldots, n\}$ denote a set of n agents and $H = \{h_1, \ldots, h_n\}$ denote a set of n houses. Each agent $i \in N$ has a preference P_i over the houses in H.[1] We denote the weak part of a preference P with R. Since P is a strict preference, aRb implies either aPb or $a = b$. For $1 \leq k \leq n$ and any $\hat{H} \subseteq H$, we denote by $r_k(P, \hat{H})$ the k-th ranked house in set $\hat{H}$ with respect to the preference P, that is, $r_k(P, \hat{H}) = a \in \hat{H}$ if $|\{b \in \hat{H} \mid bRa\}| = k$. A collection of preferences, one for each agent, is called a preference profile. We denote a preference profile $(P_1, \ldots, P_n)$ by P_N.

We now introduce a particular type of preferences known as *single-peaked* preference in the literature. A priority order $\prec$ on the set H is a strict ordering of the elements of H.[2] A preference P on H is single-peaked with respect to a priority order $\prec$ on H if preference declines as one goes farther away (with respect to $\prec$) from their top-ranked (peak) house.

Definition 5.2.1 *A preference P on H satisfies **single-peakedness** with respect to a priority order $\prec$ on H if for all $a, b \in H$, aPb whenever $[r_1(P) \prec a \prec b$ or $b \prec a \prec r_1(P)]$.*

We denote by $\mathcal{S}(\prec)$ the set of all single-peaked preferences on H with respect to $\prec$.

[1] A preference on a set S is a complete, transitive and anti-symmetric binary relation on S.

[2] Technically, a priority order is a preference. We do not call it a preference as it does not represent the same.

A multiple single-peaked domain is defined as the union of several single-peaked domains. Let $\mathbb{L}$ be a collection of priority orders. A multiple single-peaked domain with respect to $\mathbb{L}$ is defined as $\mathcal{S}(\mathbb{L}) := \cup_{\prec \in \mathbb{L}} \mathcal{S}(\prec)$. Whenever $\mathbb{L}$ is clear from the context, we drop the notation and denote $\mathcal{S}(\mathbb{L})$ by $\mathcal{S}$. Note that every single-peaked domain (with respect to one linear order) is, by definition, also a multiple single-peaked domain.

Definition 5.2.2 *A multiple single-peaked domain is non-trivial if there are at least two different linear orders, say $\prec$ and $\prec'$ in $\mathbb{L}$ such that $\mathcal{S}(\prec) \neq \mathcal{S}(\prec')$.*

For ease of presentation, we assume without loss of generality that the priority order $\prec^*$ given by $h_1 \prec^* \cdots \prec^* h_n$ belongs to $\mathbb{L}$. With slight abuse of notation, we use $h \preceq^* h_i$ to denote any house $h \in H$ such that either $h = h_i$ or $h \prec^* h_i$. Two houses $h_{\underline{\kappa}}$ and $h_{\bar{\kappa}}$ are called boundaries of $\mathbb{L}$ if for all priority orders $\prec \in \mathbb{L}$, we have $h_1 \prec h_2 \prec \cdots \prec h_{\underline{\kappa}}$, $h_{\bar{\kappa}} \prec h_{\bar{\kappa}+1} \prec \cdots \prec h_n$, and $h_{\underline{\kappa}} \prec h_x \prec h_{\bar{\kappa}}$ for all $x \in \{\underline{\kappa}+1, \ldots, \bar{\kappa}-1\}$. Note that the boundaries of $\mathbb{L}$ need not be unique. The boundaries partition the houses into three sets: the left part $H_L = \{h_1, \ldots, h_{\underline{\kappa}}\}$, the middle part $H_M = \{h_{\underline{\kappa}+1}, \ldots, h_{\bar{\kappa}-1}\}$, and the right part $H_R = \{h_{\bar{\kappa}}, \ldots, h_n\}$. Throughout the paper, we assume that $\underline{\kappa}$ and $\bar{\kappa}$ (and hence, H_L, H_M, H_R) are arbitrary but fixed.

Let us illustrate the multiple single-peaked domain with the help of a simple example.

Example 5.2.3 *Let $H = \{h_1, \ldots, h_5\}$ and $\mathbb{L} = \{\prec, \prec'\}$ where $\prec := h_1 \prec h_2 \prec h_3 \prec h_4 \prec h_5$ and $\prec' := h_1 \prec' h_2 \prec' h_4 \prec' h_3 \prec' h_5$. The boundaries in this example can be fixed to be $h_{\underline{\kappa}} = h_2$ and $h_{\bar{\kappa}} = h_5$ and the set of all multiple single-peaked preferences is given by $\mathcal{S} = \mathcal{S}(\prec) \cup \mathcal{S}(\prec')$.*

***Remark* 5.2.4** *Given $\mathbb{L}$, there may be multiple choices of $h_{\underline{\kappa}}$ and $h_{\bar{\kappa}}$, and hence multiple possible partitions into H_L, H_M, and H_R. For instance, in Example 5.2.3, one can choose $h_{\underline{\kappa}} = h_1$ instead of h_2. One can, however, define the boundary houses uniquely as follows. Let $h_{\underline{\kappa}}$ be the maximal house among those that appear in a fixed order starting from h_1 in each priority order, and let $h_{\bar{\kappa}}$ be the minimal house among those that appear in a fixed order leading to h_n in each priority order in $\mathbb{L}$. Nevertheless, allowing the boundary houses to be arbitrary but fixed yields a richer class of rules, thereby enhancing the flexibility of the model.*

5.2.2 Matching rules and their properties

A bijection $\mu : N \to H$ is called a matching. For convenience, we sometimes denote μ^{-1} by μ itself. For instance, we write $\mu(h) = i$ to mean $\mu(i) = h$. Let $\mathcal{M}$ be the set of all matchings. The houses are matched with the agents according to some matching e, which is called the initial endowment. For ease of presentation, for an endowment e we denote by $o(h)$ the owner of the house h, that is, $o(h) = i$ if and only if $e(i) = h$. Without loss of generality, we assume $e(i) = h_i$ for all $i \in N$.

A matching rule on multiple single-peaked domain is a mapping $\phi : \mathcal{S}^n \to \mathcal{M}$ that assigns a matching to each preference profile in $\mathcal{S}^n$.

We define the three desirable properties of a matching. A matching μ is individually rational if for each agent i, the house $\mu(i)$ that it allocates to i is at least as good as the house $e(i)$ that they is endowed with. A matching rule is individually rational if it assigns an individually rational matching at every profile.

Definition 5.2.5 *A matching $\mu \in \mathcal{M}$ is **individually rational (IR)** at a preference profile $P_N \in \mathcal{S}^n$ if for each $i \in N$, $\mu(i)R_ie(i)$. A matching rule $\phi : \mathcal{S}^n \to \mathcal{M}$ is **individually rational (IR)** if $\phi(P_N)$ is IR at P_N for every $P_N \in \mathcal{S}^n$.*

A matching μ is said to be Pareto optimal at a preference profile, if there is no other matching such that every agent is weakly better off and at least one agent is strictly better off. A matching rule is Pareto optimal if it assigns a Pareto optimal matching at every preference profile.

Definition 5.2.6 *A matching $\mu \in \mathcal{M}$ is **Pareto optimal (PO)** at a preference profile $P_N \in \mathcal{S}^n$ if there exists no other matching $\mu' \in \mathcal{M}$ such that*

(i) for each $i \in N$, $\mu'(i)R_i\mu(i)$, and

(ii) for some $j \in N$, $\mu'(j)P_j\mu(j)$.

*A matching rule $\phi : \mathcal{S}^n \to \mathcal{M}$ is **Pareto optimal (PO)** if $\phi(P_N)$ is PO at every $P_N \in \mathcal{S}^n$.*

A matching rule is strategy-proof if no agent can be better off by misreporting their true preference.

Definition 5.2.7 *A matching rule $\phi : \mathcal{S}^n \to \mathcal{M}$ is **strategy-proof** if for all $i \in N$, all $P_i, P_i' \in \mathcal{S}$, and all $P_{-i} \in \mathcal{S}^{n-1}$, we have $\phi_i(P_i, P_{-i})R_i\phi_i(P_i', P_{-i})$.*

5.3 C-TTC algorithm

In this section, we introduce an algorithm that is derived by combining the Crawler algorithm and the Top Trading Cycle (TTC) algorithm. We call it C-TTC algorithm. The algorithm works in several stages, and at each stage, it applies a restricted TTC on the agents in H_M, a *'left Crawler'* algorithm on the agents in H_L and a *'right Crawler'* algorithm on the agents in H_R.[3] We now proceed to provide a formal description of the algorithm.

The algorithm is defined inductively and we describe a generic stage, Stage s, of the algorithm. Consider a preference profile P_N.

Stage s. We denote the owner of a house h at Stage s by $o^s(h)$. Similarly the endowment of agent i at Stage s is denoted be $e^s(i)$. The houses remaining in Stage s are partitioned into three sets: the set of houses remaining from H_L are denoted by H^s_L, those from H_M are denoted by H^s_M, and from H_R are by H^s_R. The initial values of the concerned parameters are as follows: $H^1_L = H_L$, $H^1_M = H_M$, $H^1_R = H_R$, $o^1(h_i) = i$ and $e^1(i) = e(i)$.

We index the remaining houses at Stage s as

$$\begin{aligned} H^s_L &= \{h^s_1, \dots, h^s_{\underline{\kappa}(s)}\}, \\ H^s_M &= \{h^s_{\underline{\kappa}(s)+1}, \dots, h^s_{\bar{\kappa}(s)-1}\}, \\ H^s_R &= \{h^s_{\bar{\kappa}(s)}, \dots, h^s_{n(s)}\}, \end{aligned}$$

where

$$h^s_1 \prec^* h^s_2 \prec^* \cdots \prec^* h^s_{n(s)}.$$

Here, one or more (but not all) of H^s_L, H^s_M, H^s_R may be empty. Denote $H^s = H^s_L \cup H^s_M \cup H^s_R$.

The algorithm for Stage s works in three steps. In Step 1, we construct a directed graph, in Step 2, we determine the outcomes for Stage s, and finally, in Step 3, we determine the remaining houses for Stage $s+1$ and re-index them appropriately for Stage $s+1$.

[3] The left Crawler algorithm is defined in [2], and the right Crawler is called the 'dual' of it obtained by proceeding in the reverse order.

Step 1: Directed Graph Construction Draw one outgoing edge from each $h \in H^s$:

- If $h \in H^s_L$, draw an edge to the best house of $o^s(h)$ in $H^s_L \cup H^s_M \cup \{h^s_{\bar{\kappa}(s)}\}$.
- If $h \in H^s_R$, draw an edge to the best house of $o^s(h)$ in $H^s_R \cup H^s_M \cup \{h^s_{\underline{\kappa}(s)}\}$.
- If $h \in H^s_M$, draw an edge to the best house of $o^s(h)$ in $H^s_M \cup \{h^s_{\underline{\kappa}(s)},\ h^s_{\bar{\kappa}(s)}\}$.

Step 2 (Outcome Resolution): We decide the outcomes considering three cases based on the direction of the edge emanating from $h^s_{\underline{\kappa}(s)}$ and $h^s_{\bar{\kappa}(s)}$. In each case, we apply the left and the right crawler algorithm to the remaining houses from H_L and H_R, respectively. These algorithms are defined in [2] and [31]; nevertheless, we provide a formal description of them that is tailored to our context and included here for completeness.

Left (or, Right) Crawler : Suppose that h^s_j is the first house from the left (or, right) in H^s_L (or, H^s_R) with an edge weakly towards its left (or, right).[4] Let the edge from h^s_j point to $h^s_{\hat{j}}$. Match $o^s(h^s_j)$ with $h^s_{\hat{j}}$ and get them out of the algorithm.

Let $H^{s+1}_L = \{h^s_1, \dots, h^s_{\hat{j}-1}, h^s_{\hat{j}+1}, \dots, h^s_j, \dots, h^s_{\underline{\kappa}(s)}\}$ (or, $H^{s+1}_R = \{h^s_{\bar{\kappa}(s)}, \dots, h^s_j, \dots, h^s_{\hat{j}-1}, h^s_{\hat{j}+1}, \dots, h^s_{n(s)}\}$) denote the remaining houses in H^s_L (or, H^s_R). Define the modified owners of the houses in H^{s+1}_L (or, H^{s+1}_R) as follows:
$o^{s+1}(h^s_k) = o^s(h^s_{k-1})$ for all $\hat{j} < k \leq j$ (or, $o^{s+1}(h^s_k) = o^s(h^s_{k+1})$ for all $j \leq k < \hat{j}$),
and $o^{s+1}(h^s_k) = o^s(h^s_k)$ otherwise.
If no house points weakly towards its left (right), then no house or agent is removed, and consequently, $o^{s+1}(h^s_k) = o^s(h^s_k)$ for all $k = 1, 2, \dots, \underline{\kappa}(s)$ (or, $\bar{\kappa}(s), \dots, n(s)$).

Remark 5.3.1 *For ease of exposition, we allow the left (or, right) crawler to be applied even when H^s_L (or, H^s_R) is empty; this has no effect, as H^s_L (or, H^s_R) remains empty.*

Case 1. Suppose that $h^s_{\underline{\kappa}(s)}$ points weakly to its left or H^s_L is empty and $h^s_{\bar{\kappa}(s)}$ points weakly to its right or H^s_R is empty.

We apply the left crawler on H^s_L and the right crawler on H^s_R. For each cycle $(h^s_{j_1}, \dots, h^s_{j_k})$ in H^s_M, match $o^s(h^s_{j_l})$ with $h^s_{j_{l+1}}$ for all $l = 1, \dots, k$.[5] Remove the matched agents and houses from the algorithm and

[4] We say 'weakly towards left (or, right)' to mean either strictly to the left (or, right) or to itself.
[5] Throughout the paper, in such situations we mean $k + 1 \equiv 1$.

move to Step 3.

Case 2. Suppose that $h^s_{\underline{\kappa}(s)}$ points strictly to its right and $h^s_{\bar{\kappa}(s)}$ points strictly to its left.

We apply the left crawler on $H^s_L \setminus h^s_{\underline{\kappa}(s)}$ and the right crawler on $H^s_R \setminus h^s_{\bar{\kappa}(s)}$.

i. For each cycle $(h^s_{j_1}, \ldots, h^s_{j_k})$ involving (some of) the houses in H^s_M, match $o^s(h^s_{j_l})$ with $h^s_{j_{l+1}}$ for all $l = 1, \ldots, k$.

ii. For each cycle $(h^s_{j_1}, \ldots, h^s_{j_k})$ in $H^s_M \cup \{h^s_{\underline{\kappa}(s)}, h^s_{\bar{\kappa}(s)}\}$ involving at least one of the houses $h^s_{\underline{\kappa}(s)}$ or $h^s_{\bar{\kappa}(s)}$, define the modified owners for stage $s+1$ as $o^{s+1}(h^s_{j_{l+1}}) = o^s(h^s_{j_l})$ for all $l = 1, \ldots, k$. [6]

Keep the owners of all the other houses in $H^s_M \cup \{h^s_{\underline{\kappa}(s)}, h^s_{\bar{\kappa}(s)}\}$ unchanged. Remove the matched agents and houses from the algorithm and move to Step 3. We emphasize that a house that forms a cycle with one of the boundary houses $h^s_{\underline{\kappa}(s)}$ or $h^s_{\bar{\kappa}(s)}$ will not be removed from the algorithm.

Case 3. Suppose that $h^s_{\underline{\kappa}(s)}$ points weakly to its left (or, strictly to its right) or H^s_L is empty and $h^s_{\bar{\kappa}(s)}$ points strictly to its left (or, weakly to its right) or H^s_R is empty .

We apply the left crawler on H^s_L (or, $H^s_L \setminus \{h^s_{\underline{\kappa}(s)}\}$) and the right crawler on $H^s_R \setminus \{h^s_{\bar{\kappa}(s)}\}$ (or, H^s_R).

i. For each cycle $(h^s_{j_1}, \ldots, h^s_{j_k})$ involving only the houses in H^s_M, match $o^s(h^s_{j_l})$ with $h^s_{j_{l+1}}$ for all $l = 1, \ldots, k$.

ii. For each cycle $(h^s_{j_1}, \ldots, h^s_{j_k})$ in $H^s_M \cup \{h^s_{\underline{\kappa}(s)}, h^s_{\bar{\kappa}(s)}\}$ involving at least one of the houses $h^s_{\underline{\kappa}(s)}$ or $h^s_{\bar{\kappa}(s)}$, define the modified owners for stage $s+1$ as $o^{s+1}(h^s_{j_{l+1}}) = o^s(h^s_{j_l})$ for all $l = 1, \ldots, k$.

Keep the owners of all the other houses in $H^s_M \cup \{h^s_{\bar{\kappa}(s)}, h^s_{\bar{\kappa}(s)}\}$ unchanged. Remove the matched agents and houses from the algorithm and move to Step 3.

Step 3 (Re-indexation): Let the remaining houses from H^s_L, H^s_M, and H^s_R be denoted by H^{s+1}_L, H^{s+1}_M and H^{s+1}_R, respectively. Re-index them as $\{h^{s+1}_1, \ldots, h^{s+1}_{\underline{\kappa}(s+1)}\}$, $\{h^{s+1}_{\underline{\kappa}(s+1)+1}, \ldots, h^{s+1}_{\bar{\kappa}(s+1)-1}\}$, and $\{h^{s+1}_{\bar{\kappa}(s+1)+1}, \ldots, h^{s+1}_{n(s+1)}\}$, respectively, such that $h^{s+1}_1 \prec^* h^{s+1}_2 \prec^* \cdots \prec^* h^{s+1}_{n(s+1)}$.

This completes the description of (the generic) Stage s of the algorithm.

[6] Note that these cycles are disjoint from the ones involving (only) the houses in H^s_M.

Observation 1 *In the C-TTC algorithm, each agent's assignment moves monotonically away from their initial endowment toward their final outcome. Formally, for any agent i and any two stages* s, s', *it is impossible that* $e^s(i) \prec e(i) \prec e^{s'}(i)$ *or* $e^{s'}(i) \prec e(i) \prec e^s(i)$. *In other words, no agent ever revisits an assignment that lies on the opposite side of their initial endowment relative to their final allocation.*

It follows from the description that if no house-agent pair is matched by the left or right crawler on H_L^s and H_R^s, respectively, then, by standard logic, the TTC must form a cycle in H_M^s. Therefore, at each stage at least one agent must either leave the algorithm or change their endowment. This , together with the monotonicity property mentioned in Observation 1, implies that at least one pair of house and agent is removed at each stage, and hence, the algorithm terminates after a finite number of stages.

Remark 5.3.2 *As noted earlier in Remark 5.2.4, different choices of the boundary houses* $h_{\underline{\kappa}}$ *and* $h_{\bar{\kappa}}$ *induce different partitions* (H_L, H_M, H_R). *Since the definition of the C-TTC rule depends on these boundary houses, this gives rise to a large class of C-TTC rules.*

Note that for any $\mathbb{L}$, *one possible choice of boundary houses is such that* H_L *and* H_R *are empty and* $H_M = H$. *Under this choice, the C-TTC algorithm operates as follows. In Step 1, for each house in* H_M^s*—which in this case coincides with* H^s*—we draw an edge to the most preferred house of its owner. Since* H_L^s *and* H_R^s *are empty, Step 2 follows Case 1, and all cycles in* $H_M^s = H^s$ *are matched and removed. Step 3 simply reindexes the houses for the next stage. Therefore, under this choice of boundary houses, the C-TTC rule reduces to the TTC rule.*

We illustrate the workings of the C-TTC algorithm with the help of the following example.

Example 5.3.3 *Consider* $N = \{1, \ldots, 7\}$ *and* $H_L = \{h_1, h_2\}$, $H_M = \{h_3, h_4, h_5\}$, $H_R = \{h_6, h_7\}$ *and a preference profile given as follows:*

Table 5.3.1

P_1	P_2	P_3	P_4	P_5	P_6	P_7
h_5	h_1	h_7	h_5	h_6	h_1	h_2
h_2	h_2	h_6	h_3	h_7	h_2	h_5
h_3	h_5	h_3	h_4	h_5	h_3	h_1
h_4	h_3	h_4	h_2	h_4	h_4	h_4
h_6	h_4	h_5	h_6	h_3	h_5	h_3
h_7	h_6	h_2	h_1	h_2	h_6	h_6
h_1	h_7	h_1	h_7	h_1	h_7	h_7

Table 5.3.2: Preferences of agents

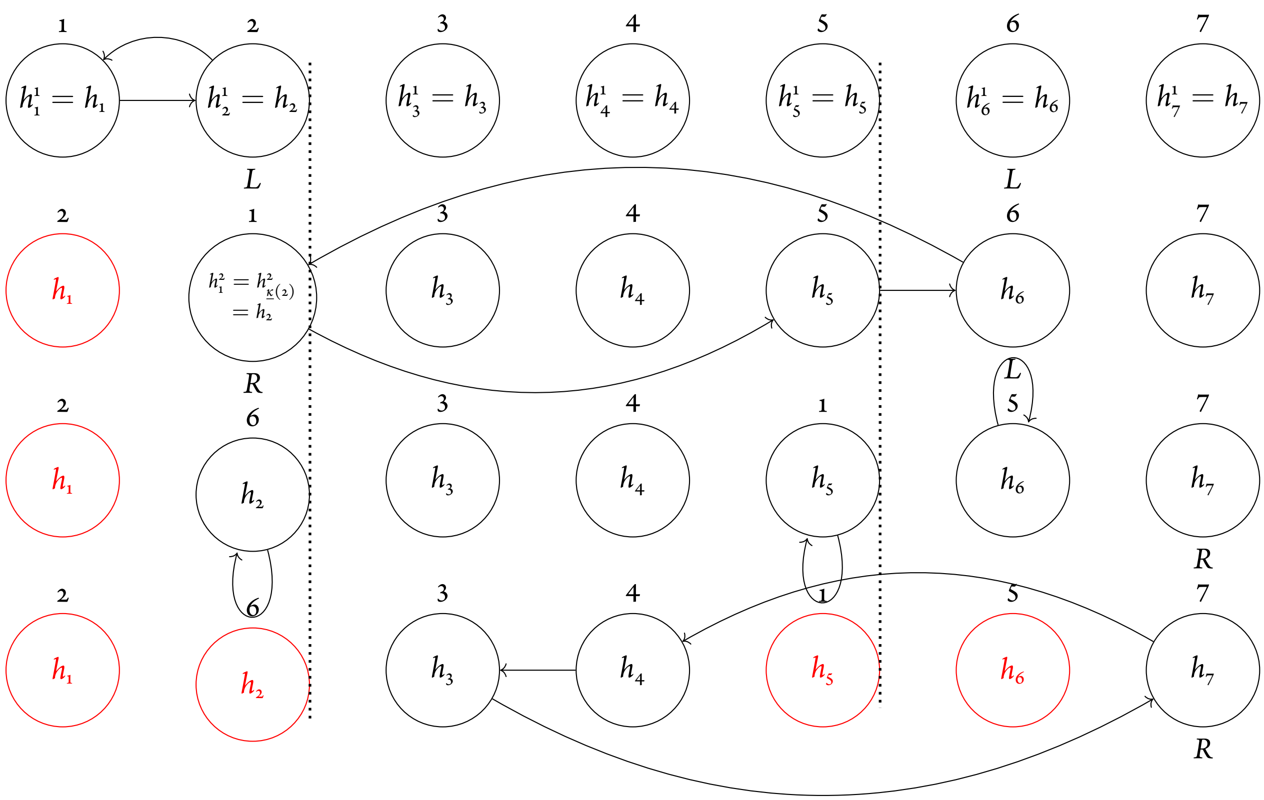

Figure 5.3.1: Illustration of the successive stages of the C-TTC algorithm.

The figure illustrates the C-TTC algorithm on the set of seven houses. Each row represents a stage of the algorithm. Each circle represents a house, with the occupant's agent listed above it. Red-shaded nodes indicate houses (and their agents) that have been removed from the market at that stage. For clarity, we display only those directed edges corresponding to agents whose endowments either shift or are eliminated in the current stage. The L and R markings represent the direction at which the boundary agents point at that stage, which determines the respective case to use in the outcome resolution step.

At stage 1, agents in H_L^1 point to their best house in $H_L^1 \cup H_M^1 \cup h^1_{\bar{\kappa}(1)} = \{h_1, \ldots, h_6\}$. Thus agents 1 and 2 point to h_5 and h_1, respectively. Agents in H_R^1 point to their best house in $H_R^1 \cup H_M^1 \cup h^1_{\underline{\kappa}(1)} = \{h_2, \ldots, h_7\}$. Thus agents 6 and 7 point to houses h_2 and h_2, respectively. Agents in H_M^1 point to their best house in $H_M^1 \cup h^1_{\underline{\kappa}(1)} \cup h^1_{\bar{\kappa}(1)} = \{h_2, \ldots, h_6\}$. Thus agent 3, 4 and 5 point to houses h_6, h_5 and h_6, respectively.

Since $h^1_{\underline{\kappa}(1)} \equiv h_2$ weakly points to their left and $h^1_{\bar{\kappa}(1)} \equiv h_6$ points strictly to their left, we apply left crawler on H_L^1 and right crawler on $H_R^1 \setminus \{h^s_{\bar{\kappa}(s)}\}$ (according to Case 3 of outcome resolution step). Thus agent 2, being the first agent from the left in H_L^1 who points weakly to their left, is matched with their best house i.e. h_1 and agent 1 who is currently occupying house h_1 is shifted one house to the right i.e. h_2. Since all agents in H_R^1 point to their left, there is no agent who is the first from the right in H_R^1 who points weakly to their right. Thus no changes are made in H_R^1. We further note that no cycle is formed involving owners of the houses in the set $H_M^1 \cup h^1_{\underline{\kappa}(1)} \cup h^1_{\bar{\kappa}(1)}$. We end the stage 1 with agent 2 leaving with house h_1.

At stage 2, we again construct the directed graph as before. Observe that there are no agents in H_L^2 who point weakly to their left and no agents in H_R^2 who point weakly to their right. Thus we apply Case 2 of the outcome resolution step. A cycle is formed involving agents 1, 5 and 6. Since this cycle is not restricted to H_M^2 (as it involves the boundary houses), we simply exchange the houses and no one leaves.

At stage 3, $h^3_{\underline{\kappa}(3)} \equiv h_2$ points weakly to their left and $h^3_{\bar{\kappa}(3)} \equiv h_6$ points weakly to their right. Thus we apply Case 1 of the outcome resolution step. Agent 6 who currently occupies h_2 is the first agent in H_L^3 pointing weakly to their left. So they leave the market by pointing to their best i.e. h_2. Agent 5 who occupies h_6 is the first agent from the right in H_R^3 pointing weakly to their right. So they leaves the market by pointing to their best i.e. h_6. The only cycle formed in H_M^3 is with agent 1 who points to their current house h_5. Since this cycle involves only agents in H_M^3, agent 1 leaves with house h_5.

Finally at stage 4, the set H_L^4 is empty and $h^s_{\bar{\kappa}(4)} \equiv h_7$ points strictly to their left. Hence we apply Case 3 of the *outcome resolution step. The remaining agents* 3, 4 *and* 7 *who form a cycle. Since this cycle involves houses outside* H_M^4*, we exchange the houses and do not let them leave at this stage. However they will leave in the following stage, when all of them will point towards themselves in the similar way agents* 6, 1, 5 *left the algorithm at stage* 3.

Observation 2 *At any preference profile* P_N*, suppose agent* $i \in N$ *with* $e(i) \notin H_M$ *is matched to some house* $h \in H_M$ *at Stage* s*. Then there must exist an earlier Stage* $s^* < s$ *at which* i *occupied a boundary house (either* $h^{s^*}_{\underline{\kappa}(s^*)}$ *or* $h^{s^*}_{\bar{\kappa}(s^*)}$*). Since no cycle involving* i *formed up through Stage* $s-2$*, they first participated in a cycle at Stage* $s-1$ *by pointing to* h*. Because that cycle included at least one boundary house,* i *did not exit immediately at Stage* $s-1$*; instead they moved to* h *and remained in the algorithm. Finally, at Stage* s*,* i *forms a self-cycle by pointing to their current house* $h \in H_M^s$ *and exits the algorithm with* h*.*

We present the main theorem of this section which states that the C-TTC algorithm satisfies the three desirable properties of IR, PO and strategy-proofness. For clarity, we explicitly include $\mathbb{L}$ in the notation for both the domain and the C-TTC algorithm in the statement of the theorem.

Theorem 5.3.4 *The matching rule* $\bar{\phi} : \mathcal{S}^n(\mathbb{L}) \rightarrow \mathcal{M}$ *given by the C-TTC(*$\mathbb{L}$*) algorithm satisfies individual rationality, Pareto optimality and strategy-proofness on the domain* $\mathcal{S}^n(\mathbb{L})$*.*

We relegate the proof of Theorem 5.3.4 to Appendix .1.

Remark 5.3.5 *Theorem 5.3.4 shows that the C-TTC algorithm satisfies individual rationality, Pareto optimality, and strategy-proofness on the domain* $\mathcal{S}^n(\mathbb{L})$*. In addition to these properties, the literature often considers another desirable property, namely core stability. Although the core is not the primary focus of this paper, we briefly discuss it for comparison: the TTC rule always selects an allocation in the core, whereas the crawler mechanism does not, in general, do so. Since the C-TTC algorithm is closely related to both, it is natural to ask whether its outcome always lies in the core.*

[24] show that, in the Shapley–Scarf housing market, the TTC outcome is the unique core allocation. The following example presents a preference profile for which the C-TTC outcome differs from the TTC outcome, thereby demonstrating that the C-TTC algorithm does not, in general, select an allocation in the core.

Example 5.3.6 *Consider* $N = \{1, \ldots, 7\}$ *and* $H_L = \{h_1, h_2\}$*,* $H_M = \{h_3, h_4\}$*,* $H_R = \{h_5, h_6, h_7\}$ *and a preference profile given as follows:*

Table 5.3.3

P_1	P_2	P_3	P_4	P_5	P_6	P_7
h_6	h_6	h_6	h_6	h_6	h_1	h_6
h_7	h_5	h_5	h_5	h_5	h_2	h_5
h_5	h_4	h_4	h_4	h_4	h_3	h_4
h_4	h_3	h_3	h_3	h_3	h_4	h_3
h_3	h_2	h_2	h_2	h_2	h_5	h_2
h_2	h_1	h_1	h_1	h_1	h_6	h_1
h_1	h_7	h_7	h_7	h_7	h_7	h_7

Table 5.3.4: Preferences of agents

The TTC outcome at P_N is $(h_6, h_2, h_3, h_4, h_5, h_1, h_7)$ and the C-TTC outcome at P_N is $(h_7, h_5, h_3, h_4, h_6, h_1, h_2)$

5.4 Obvious Strategy-proofness

Strategy-proofness guaranties that no agent can gain by misreporting their preferences—truth-telling is weakly dominant—yet this property can be opaque to participants who may not fully grasp the mechanism. To address this, Li [13] introduces the stronger criterion of *obvious strategy-proofness*, which requires that incentive compatibility be immediately apparent even to agents with bounded reasoning. Their experiments demonstrate starkly different behavior when a mechanism satisfies this more transparent form of strategy-proofness.

Building on this idea, we require matching rules to satisfy obvious strategy-proofness in addition to individual rationality and Pareto optimality. Under these tighter constraints, the universe of permissible mechanisms shrinks dramatically compared to the standard strategy-proof setting. In the next section, we review the extensive form framework and define obvious dominance and obvious strategy-proofness formally.

Definition 5.4.1 *An **extensive form structure** is defined as the tuple $\Xi = \langle T, N, \gamma \rangle$ where*

- *$T = (V, E)$ is a directed root tree with root at $v_0 \in V$ and the set of terminal nodes as $Z \subset V$,*
- *N is the set of players,*
- *$\gamma : V \setminus Z \to N$ is a function that specifies which agent in N moves at each non-terminal node.*

For $i \in N$, let V_i be the nodes of T where i is assigned to take an action, that is, $\gamma(v_i) = i$ for all $v_i \in V_i$. The outgoing edges from a node v is denoted by $O(v)$. Given an extensive form structure Ξ, an action of agent $i \in N$ is a mapping $\underset{\sim}{a}_i : V_i \to E$ such that $\underset{\sim}{a}_i(v_i) \in O(v_i)$ for all $v_i \in V_i$. We denote by $\underset{\sim}{A}_i$ the set of all actions of agent i, and by $\underset{\sim}{A}_N$ the set of all action profiles $\prod_{i \in N} \underset{\sim}{A}_i$ of all the agents. A strategy of an agent $i \in N$ is a mapping $s_i : \mathcal{S}_i \to \underset{\sim}{A}_i$. The strategy profile of agents in N at a preference profile is denoted by $s_N = \prod_{i \in N} s_i$. A mechanism is a mapping $m : \underset{\sim}{A}_N \to \mathcal{M}$.

For $v \in V$, we denote by T_v the restriction of the tree T rooted at v, and by Ξ_v the restriction of Ξ on T_v. We call Ξ_v the sub-game extensive form structure rooted at v. Similarly, we denote by s_i^v, s_N^v, and m^v, the restrictions of s_i, s_N, and m, on T^v.

Definition 5.4.2 *A strategy s_i is **obviously dominant** for agent $i \in N$ in a mechanism m, if for each $v_i \in V_i$, each $P_N \in \mathcal{S}^n$, every $\bar{s}_i$ such that $s_i(P_i)(v_i) \neq \bar{s}_i(P_i)(v_i)$, and all s_{-i}, we have $m^{v_i}(s_N^{v_i}(P_N)) R_i m^{v_i}(\bar{s}_i^{v_i}(P_i), s_{-i}^{v_i}(P_{-i}))$.*

Definition 5.4.3 *A matching rule $\phi : \mathcal{S}^n \to \mathcal{M}$ is said to be **obviously strategy-proof** if there exists a mechanism m and a strategy profile s_N such that*

(i) s_i is obviously dominant for each agent $i \in N$ and

(ii) $\phi(P_N) = m(s_N(P_N))$ for all $P_N \in \mathcal{S}^n$.

We now present the second result of our paper. It says that on a non-trivial multiple single-peaked domain (that is, there are at least two different linear orders in $\mathbb{L}$), no matching rule satisfies IR, PO, and obvious strategy-proofness.

Theorem 5.4.4 *There is no matching rule on a non-trivial multiple single peaked domain satisfying IR, PO, and obvious strategy-proofness.*

We relegate the proof of Theorem 5.4.4 to Appendix .2.

Appendix

.1 Proof of Theorem 5.3.4

We first prove a few lemmas establishing crucial properties of the C-TTC algorithm.

Lemma .1.1 *For any profile $P_N \in \mathcal{S}^n$ and any agent $i \in N$, if i is matched with a house $h \in H$ at Stage s of the C-TTC algorithm, then h is the most preferred house among the houses remaining at Stage s according to P_i.*

Proof: Suppose agent i gets matched at Stage s. We distinguish the following cases.

Case 1: Suppose $i \in H^s_L$. By the definition of the C-TTC algorithm, i must have pointed to themselves or to their left at Stage s. By the domain assumption $r_1\left(P_i, H^s_L \cup H^s_M \cup \{h^s_{\underline{\kappa}(s)}, h^s_{\bar{\kappa}(s)}\}\right) \in H^s_L$ implies that $r_1(P_i, H^s) \in H^s_L$ (since if $r_1(P_i, H^s) \in H^s_R$ then $H_L \prec^* H_M \prec^* H_R$ and single-peakedness of the domain imply that $r_1\left(P_i, H^s_L \cup H^s_M \cup \{h^s_{\underline{\kappa}(s)}, h^s_{\bar{\kappa}(s)}\}\right)$ must be $h^s_{\bar{\kappa}(s)}$). Since i is matched with $r_1\left(P_i, H^s_L \cup H^s_M \cup \{h^s_{\underline{\kappa}(s)}, h^s_{\bar{\kappa}(s)}\}\right) \in H^s_L$ at Stage s, this implies that i gets their most preferred house in the remaining houses. A similar logic applies for the case when $i \in H^s_R$.

Case 2: Suppose $i \in H^s_M$. By the definition of the C-TTC algorithm, i must have pointed to their best house in $H^s_M \cup \{h^s_{\underline{\kappa}(s)}, h^s_{\bar{\kappa}(s)}\}$. Since i gets matched at Stage s, by the definition of the C-TTC algorithm, i's most preferred house in $H^s_M \cup \{h^s_{\underline{\kappa}(s)}, h^s_{\bar{\kappa}(s)}\}$ is in H^s_M. By the domain assumption, $r_1\left(P_i, H^s_M \cup \{h^s_{\underline{\kappa}(s)}, h^s_{\bar{\kappa}(s)}\}\right) \in H^s_M$ implies that $r_1(P_i, H^s) \in H^s_M$ (if $r_1(P_i, H^s) \in H^s_L$ or H^s_R, then $H_L \prec^* H_M \prec^* H_R$ and single-peakedness of the domain imply that $r_1(P_i, H^s) = h^s_{\underline{\kappa}(s)}$ or $h^s_{\bar{\kappa}(s)}$).Since i is matched with $r_1(P_i, H^s_M \cup \{h^s_{\underline{\kappa}(s)}, h^s_{\bar{\kappa}(s)}\}) \in H^s_M$ at Stage s, this implies that i gets their most preferred house in the remaining houses.

This completes the proof of Lemma .1.1. ■

Lemma .1.2 *Consider a profile P_N and suppose that $e^s(i) \in H^s_L \cup H^s_R$ for some Stage s. If an agent i points weakly towards left (or, right) of $e^s(i)$, then $\bar{\phi}_i(P_N) \preceq^* e^s(i)$ (or, $e^s(i) \preceq^* \bar{\phi}_i(P_N)$).*

Proof:

Claim .1.1 *Consider a profile P_N and suppose that for some Stage s, $e^s(i) \neq e^{s+1}(i) \in H^{s+1}_M$. Then, $\bar{\phi}_i(P_N) = e^{s+1}(i)$.*

Proof: [Proof of the claim] From the definition of the C-TTC algorithm, it is evident that $e^s(i) \in H^s_M \cup \{h^s_{\underline{\kappa}(s)}, h^s_{\bar{\kappa}(s)}\}$. Since $e^{s+1}(i) \neq e^s(i)$, agent i must have formed a cycle at Stage s by pointing at $e^{s+1}(i)$ which implies that $r_1\big(P_i, H^s_M \cup \{h^s_{\underline{\kappa}(s)}, h^s_{\bar{\kappa}(s)}\}\big) = e^{s+1}(i)$. On the other hand, we know that if $P_i \in \mathcal{S}$, then $r_1(P_i, H^s) = r_1(P_i, H^s_M \cup \{h^s_{\underline{\kappa}(s)}, h^s_{\bar{\kappa}(s)}\})$. Hence at stage $s+1$, agent i points towards their own house $e^{s+1}(i)$ and leaves the C-TTC algorithm. Therefore we have $\phi_i(P_N) = e^{s+1}(i)$.

■

We provide the proof for the case i points towards left; the proof for the case where i points towards right is logically equivalent.

We distinguish a few cases.

Case 1. $e^s(i) \in H^s_L$ Agent i is part of Left Crawler. If i is the first left pointing agent in H^s_L then $e^{s+1}(i) = \phi_i(P_N) \preceq^* e^s(i)$. Else $e^{s+1}(i) = e^s(i)$ and $r_1(P_i, H^{s+1}) \preceq^* e^{s+1}(i)$ (since $P_i \in \mathcal{S}$). Hence i points to the left at stage $s+1$. Continuing this logic for every stage until agent i leaves the C-TTC algorithm we conclude the proof for this case.

Case 2. $e^s(i) \in H^s_R$

Case 2.1. $e^s(i) = h^s_{\bar{\kappa}(s)}$. If it does not form a cycle, then $e^{s+1}(i) = e^s(i)$, and we apply the same logic to the next stage. Otherwise, if it forms a cycle pointing to some house in H^s_M, then by claim .1.1, we have $\phi_i(P_N) = e^{s+1}(i)$. Else it forms a cycle with $h^s_{\underline{\kappa}(s)}$. Then, in the next stage, $e^{s+1}(i) = h^{s+1}_{\underline{\kappa}(s+1)} \in H^{s+1}_L$, and we apply Case 1 to conclude the proof.

Case 2.2. $e^s(i) \neq h^s_{\bar{\kappa}(s)}$. Then agent i is part of Right Crawler. If the first right pointing agent in H^s_R is to the left of i and points to any house h which is weakly to the right of $e^s(i)$ i.e., $e^s(i) \preceq^* h$, then $e^{s+1}(i) \prec^* e^s(i)$ and $r_1(P_i, H^{s+1}) \preceq^* e^{s+1}(i)$. Else we have $e^{s+1}(i) = e^s(i)$ and $r_1(P_i, H^{s+1} \preceq^* e^{s+1}(i)$. Hence $e^{s+1}(i) \in H^{s+1}_R$ and agent i continues pointing towards left at stage $s+1$ and we apply Case 2 to conclude the proof.

■

Lemma .1.3 *For every profile P_N and every Stage s of the C-TTC algorithm, we have $\phi_i(P_N) R_i e^s(i)$.*

Proof: We distinguish the following cases based on the position of $e^s(i)$.

Case 1. Suppose $e^s(i) \in H^s_L \cup H^s_R$. Assume further that agent i points to the left of $e^s(i)$. Then from the proof of Lemma .1.2, we have either $e^{s+1}(i) = r_1(P_i, H^S)$ (when agent i is the first left pointing agent in H^s_L

or when agent i forms a cycle by pointing at $e^{s+1}(i) \in H^s_M$),or $e^{s+1}(i) = h^s_{\underline{\kappa}(s)}$, or $e^{s+1}(i) = e^s(i)$, or $e^{s+1}(i)$ as the immediate left neighboring house of $e^s(i)$. In the first instance, we have $\phi_i(P_N) = e^{s+1}(i)$. In the second instance, $e^s(i) = h^s_{\overline{\kappa}(s)}$ and $r_1(P_i, H^s) \preceq^* r_1(P_i, H^s_M \cup \{h^s_{\underline{\kappa}(s)}, h^s_{\overline{\kappa}(s)}\}) = e^{s+1}(i)$. In the last two instances, we have $r_1(P_i, H^s) \preceq^* e^{s+1}(i) \preceq^* e^s(i)$. Since this holds true for all stages until agent i leaves the C-TTC algorithm, we have $\phi_i(P_N) R_i e^s(i)$. Similar logic applies when agent i points to the right of $e^s(i)$.

Case 2. Suppose $e^s(i) \in H^s_M$. Note that by the definition of the C-TTC algorithm, $e^{s+1}(i) \in H^s_M \cup \{h^s_{\underline{\kappa}(s)}, h^s_{\overline{\kappa}(s)}\}$. Suppose that $e^{s+1}(i) \in H^s_M$. If $e^{s+1}(i) = e^s(i)$, then apply Case 2 for $e^{s+1}(i)$. If $e^{s+1}(i) \neq e^s(i)$, then from Claim .1.1, we have $\phi_i(P_N) = e^{s+1}(i) = r_1(P_i, H^s_M \cup \{h^s_{\underline{\kappa}(s)}, h^s_{\overline{\kappa}(s)}\})$. If $e^{s+1}(i) \in \{h^s_{\underline{\kappa}(s)}, h^s_{\overline{\kappa}(s)}\}$, then by applying the logic for Case 1, we have $\phi_i(P_N) R_i e^{s+1}(i)$. This along with the fact that $r_1(P_i, H^s_M \cup \{h^s_{\underline{\kappa}(s)}, h^s_{\overline{\kappa}(s)}\}) = e^{s+1}(i)$ completes the proof. ■

Proof: We are now ready to proceed with the proof of Theorem 5.3.4. We begin with proving strategy-proofness.

Strategy-proofness : Let ϕ denote the outcome of the C-TTC algorithm. Consider an agent $i \in N$ and a preference profile $P_N \in \mathcal{S}^n$. We need to show that $\phi_i(P_i, P_{-i}) R_i \phi_i(P'_i, P_{-i})$ for all $P'_i \in \mathcal{S}$. Suppose that agent i gets matched with $\phi_i(P_i, P_{-i})$ at Stage s of the C-TTC algorithm.

Suppose $e^t(i, P_i)$ denote the endowment of agent i at stage t when agent i reports P_i, given that all agents other than i report P_{-i}. Let

$$e^u(i, P_i) = e^u(i, P'_i) \text{ for all } u \leq t^*, \text{ and } e^{t^*+1}(i, P_i) \neq e^{t^*+1}(i, P'_i). \tag{1}$$

In other words, under P'_i, agent i's endowment remains the same as their endowment under P_i until stage $t^* < s$ and at stage $t^* + 1$, their endowment is different for the first time.

We now consider the following cases based on the location of $e^{t^*}(i, P_i)$.

Case 1. $e^{t^*}(i, P_i) \in H^{t^*}_L \setminus h^{t^*}_{\underline{\kappa}(t^*)}$.

Let j be the first left-pointing agent at Stage t^* at profile (P_i, P_{-i}). If $j = i$, then i receives the best house in the set of remaining houses at Stage t^*, and hence cannot be strictly better off by any misreport. If $j < i$, then, by the definition of the C-TTC algorithm, no matter what agent i does, their allocation in the next Stage cannot change, which is a contradiction to our assumption on t^*.

Assume that $j > i$ at the profile (P_i, P_{-i}). This, in particular, implies that i points to their right at Stage

t^* according to the preference P_i. We consider two further subcases.

Case 1.1. If i continues to point towards their right at Stage t^* according to the preference P'_i, then by the definition of the C-TTC algorithm, i's endowment will change in exactly the same way in both (P_i, P_{-i}) and (P'_i, P_{-i}). More precisely, either $e^{t^*+1}(i, P_i) = e^{t^*}(i, P_i)$ and $e^{t^*+1}(i, P'_i) = e^{t^*}(i, P'_i)$, or $e^{t^*+1}(i, P_i) = e^{t^*}(i+1, P_i)$ and $e^{t^*+1}(i, P'_i) = e^{t^*}(i+1, P'_i)$. In each of the cases, we have $e^{t^*+1}(i, P_i) = e^{t^*+1}(i, P'_i)$, which is a contradiction to our assumption of t^*.

Case 1.2. Assume finally that i points weakly towards their left according to P'_i. Recall that by the assumption of this case, i points towards their right according to P_i. Since the first left-pointing agent j according to P_N is on the right of i, i becomes the first left-pointing agent according to (P'_i, P_{-i}). Therefore, by the definition of the C-TTC algorithm, we have $e^{t^*+1}(i, P'_i) \prec^* e^{t^*}(i, P'_i)$. However, since i points towards their right at Stage t^* according to P_i, it must be that the best house of agent i in the set of remaining houses in Stage t^* is on the right of $e^{t^*}(i, P_i)$. By the single-peaked assumption on the preferences, this implies $e^{t^*}(i, P'_i) R_i \phi(P'_i, P_{-i})$. However, by Lemma .1.3, $\phi(P_i, P_{-i}) R_i e^{t^*}(i, P_i) = e^{t^*}(i, P'_i)$. Combining these two facts, we obtain that $\phi(P_i, P_{-i}) R_i \phi(P'_i, P_{-i})$.

Case 2. $e^{t^*}(i, P_i) = h^{t^*}_{\underline{\kappa}(t^*)}$.

We consider two further subcases.

Case 2.1. Suppose agent i points to their left according to P_i. If i is the first left-pointing agent at Stage t^* according to P_N, then i gets the best house in the set of available houses at Stage t^*, and hence they cannot manipulate. Suppose i is not the first left-pointing agent at Stage t^* according to P_N. Since i's endowments are different at Stage t^*+1 according to P_i and P'_i, it must be that i points towards their right according to P'_i at Stage t^*. Using similar logic as for Case 1.2, it follows that agent i cannot be better off by manipulating.

Case 2.2. Suppose agent i points towards their right according to P_i. If i points towards their left according to P'_i, then by using a logic similar to that of Case 1.2, i cannot be better off off by manipulating. Suppose i points towards their right according to P'_i. We further distinguish two cases based on whether i forms a cycle under P_i or not.

Case 2.2.1. Suppose i forms a cycle under P_i by pointing to a house $h \in H^{t^*}_M \cup h^{t^*}_{\bar{\kappa}(t^*)}$.

Case 2.2.1.a. If $h \in H^{t^*}_M$, then from Claim .1.1, we know that $\phi_i(P_i, P_{-i}) = h$. Therefore, i gets the best house available at H^{t^*}, and thus cannot be better off by misreporting to P'_i.

Case 2.2.1.b. Suppose $h = h^{t^*}_{\bar{\kappa}(t^*)}$.

If $h = h^{t^*}_{\bar{\kappa}(t^*)}$, then by the definition of the multiple single-peaked domain, we must have $h' \prec^* h^{t^*}_{\bar{\kappa}(t^*)} \preceq^* r_1(P_i, H^{t^*})$ for any $h' \in H^{t^*}_M$. Therefore, if i forms a cycle under P'_i pointing to some house $h' \in H^{t^*}_M$, then from Claim .1.1, we have $\phi_i(P'_i, P_{-i}) = h'$. Note that since agent i forms a cycle by pointing to house $h^{t^*}_{\bar{\kappa}(t^*)}$, we must have $e^{t^*+1}(i, P_i) = h^{t^*}_{\bar{\kappa}(t^*)}$. Combining this with Lemma .1.3, we have $\phi_i(P_i, P_{-i}) R_i h^{t^*}_{\bar{\kappa}(t^*)} . R_i \phi_i(P'_i, P_{-i})$. Suppose i does not form a cycle under P'_i at stage t^*. Then at some later stage $\hat{t} > t^*$, agent i must form a cycle. Using similar logic as earlier, either $e^{\hat{t}}(i, P'_i) \in H^{\hat{t}}_M$ or $e^{\hat{t}}(i, P'_i) = h^{\hat{t}}_{\bar{\kappa}(\hat{t})}$. In the former case, we have $\phi_i(P'_i, P_{-i}) \in H_M$, which implies that $\phi_i(P_i, P_{-i}) R_i \phi_i(P'_i, P_{-i})$, while in the later case, we have $\phi_i(P'_i, P_{-i}) = \phi_i(P_i, P_{-i})$.

Case 2.2.2. Suppose i does not form a cycle according to P_i. Suppose i does not form a cycle according to P'_i as well. Then, by the definition of the C-TTC algorithm, we have $e^{t^*}(i, P'_i) = e^{t^*+1}(i, P'_i)$. Because i also does not form a cycle under P_i, this implies $e^{t^*}(i, P_i) = e^{t^*+1}(i, P_i)$. However, since $e^{t^*}(i, P_i) = e^{t^*}(i, P'_i)$, this contradicts our assumption that $e^{t^*+1}(i, P_i) \neq e^{t^*+1}(i, P'_i)$ as given in Equation 1. Therefore, it must be that i forms a cycle according to P'_i. Let us denote the cycle as $(h^{t^*}_{i_1}, \ldots, h^{t^*}_{i_k})$. Assume without loss of generality that $i_1 = i$. Therefore, by the definition of the algorithm, we have $e^{t^*+1}(i, P'_i) = h^{t^*}_{i_2}$.

Let $t' > t^*$ be the first stage after stage t^* along the C-TTC algorithm when agent i's endowment changes under P_i, that is, $e^{t'+1}(i, P_i) \neq e^{t'}(i, P_i)$ and $e^t(i, P_i) = e^{t+1}(i, P_i)$ for all $t^* \leq t < t' - 1$.

Claim .1.2 *Each of the houses $(h^{t^*}_{i_1}, \ldots, h^{t^*}_{i_k})$ will stay in the C-TTC algorithm at the stage t'.*

Proof:[Proof of the claim] Since i_k points to $e^{t^*}(i_1)$ at stage t^*, i_k will continue to point $e^{t^*}(i_1)$ as long as $e^{t^*}(i_1)$ stays in the algorithm. By the definition of t', this implies i_k will continue to point $e^{t^*}(i_1)$ until stage t'. Consequently, i_k will remain in the algorithm until at least stage t'. Again, since i_{k-1} points to $e^{t^*}(i_k)$, using a similar logic, i_{k-1} will continue pointing to $e^{t^*}(i_1)$ until at least stage t' and will remain in the algorithm until that stage. Continuing in this manner, all the houses involved in the cycle $(h^{t^*}_{i_1}, \ldots, h^{t^*}_{i_k})$ and their respective owners $i_1, \ldots, i_k$ will stay in the algorithm until stage t'. Therefore, agent i can still get the same house $h^{t^*}_{i_2}$ under P_i at the later stage t'. ■

We now distinguish two cases based on the location of $e^{t^*+1}(i, P'_i)$. By our assumption of Case 2.2, either $e^{t^*+1}(i, P'_i) \in H^{t^*}_M$ or $e^{t^*+1}(i, P'_i) = h^{t^*}_{\bar{\kappa}(t^*)}$. Suppose $e^{t^*+1}(i, P'_i) \in H^{t^*}_M$. Then by Claim .1.1, we have

$\phi_i(P'_i, P_{-i}) = e^{t^*+1}(i, P'_i)$. Since $e^{t^*+1}(i, P'_i)$ is also available to agent i at stage t' under P_i, we have $e^{t'+1}(i, P_i) R_i e^{t^*+1}(i, P'_i)$. From Lemma .1.3, we know that $\phi_i(P_N) R_i e^s(i)$ for every stage s and every profile P_N, and hence we have $\phi_i(P_i, P_{-i}) R_i e^{t^*}(i, P_i)$. Combining all these facts, we have $\phi_i(P_i, P_{-i}) R_i \phi_i(P'_i, P_{-i})$.

Now suppose $e^{t^*+1}(i, P'_i) = h^{t^*}_{\bar{\kappa}(t^*)}$. From the definition of the C-TTC algorithm, it follows that under P'_i agent i will remain at $h^{t^*}_{\bar{\kappa}(t^*)}$ until the stage, say $\hat{\bar{t}}$, when agent i is the first weakly right pointing agent in H_R. We further distinguish two cases based on the location of the house i points to under P_i at stage t'.

Case 2.2.2.a Suppose under P_i, if at stage t' agent i forms a cycle by pointing to any house $h \in H^{t'}_M$. Then by Claim .1.1, we have $\phi_i(P_i, P_{-i}) = h$. As we have argued in Claim .1.2, $h^{t^*}_{\bar{\kappa}(t^*)}$ must be available to agent i at stage t'. Because i points to a different house h, we have $h\, P_i\, h^{t^*}_{\bar{\kappa}(t^*)}$. Furthermore, because $\phi_i(P_i, P_{-i}) = h$, we have $\phi_i(P_i, P_{-i})\, P_i\, h^{t^*}_{\bar{\kappa}(t^*)}$. By the structure of the single-peaked domain, we have $h^{t^*}_{\bar{\kappa}(t^*)}\, R_i\, h'$ for any $h' \in H^{t'}_R$. Combining, we have,

$$\phi_i(P_i, P_{-i})\, P_i\, h^{t^*}_{\bar{\kappa}(t^*)}\, R_i\, h' \text{ for any } h' \in H^{t'}_R \tag{2}$$

Because $e^{t^*}(i, P'_i) = h^{t^*}_{\underline{\kappa}(t^*)}$ and $e^{t^*+1}(i, P'_i) = h^{t^*}_{\bar{\kappa}(t^*)}$, it must be that $r_1(P'_i, H^{t^*})$ belongs to $H^{t^*}_R$. Therefore, under P'_i, i must point weakly to the right of $h^{t^*}_{\bar{\kappa}(t^*)}$ in stage $t^* + 1$. By Lemma .1.2, this means $\phi(P'_i, P_{-i})$ must be weakly to the right of $h^{t^*}_{\bar{\kappa}(t^*)}$, that is, $\phi(P'_i, P_{-i}) \in H^{t^*}_R$. By Equation 2, this implies $\phi_i(P_i, P_{-i}) P_i \phi_i(P'_i, P_{-i})$.

Case 2.2.2.b Suppose at stage t' agent i chooses to form a cycle by pointing to the house $h^{t^*}_{\bar{\kappa}(t^*)}$. Then, as we have mentioned earlier in Case 2.2.2 for P'_i, agent i will remain at $h^{t^*}_{\bar{\kappa}(t^*)}$ under P_i until the stage when agent i is the first weakly right pointing agent in H_R. By the definition of the algorithm, this can only happen when all the agents in $H_R \setminus \{h^{t^*}_{\bar{\kappa}(t^*)}\}$ are pointing strictly to their left. Let that stage be denoted by $\bar{t}$. Recall that at stage $\hat{\bar{t}}$, agent i is the first weakly right pointing agent in H_R under P'_i.

Claim .1.3 *We claim that $H^{\hat{\bar{t}}}_R = H^{\bar{t}}_R$.*

Assume for contradiction that $H^{\hat{\bar{t}}}_R \neq H^{\bar{t}}_R$. Note that $e^{t^*}(i, P_i) = h^{t^*}_{\underline{\kappa}(t^*)}$ and i points weakly to the right at stage t^* under P_i.

We argue that $e^t(i, P_i) \neq h^{t^*}_{\bar{\kappa}(t^*)}$ at any $t < t^*$. Suppose to the contrary that $e^t(i, P_i) = h^{t^*}_{\bar{\kappa}(t^*)}$ at some stage $t < t^*$. Then i must have pointed weakly towards the left of $h^{t^*}_{\underline{\kappa}(t^*)}$ at some stage t'' where $t < t'' \leq t^*$

to arrive at $h^{t^*}_{\underline{\kappa}(t^*)}$ at stage t^*. This implies that the most preferred house of i under P_i in the set of available houses $H^{t'}$ in stage t' is weakly on the left of $h^{t^*}_{\underline{\kappa}(t^*)}$. By the structure of single-peaked domain, this means $h^{t^*}_{\underline{\kappa}(t^*)} \, P_i \, h^{t^*}_{\overline{\kappa}(t^*)}$. This contradicts the fact that $e^{t^*}(i, P_i) = h^{t^*}_{\underline{\kappa}(t^*)}$ and i points to the $h^{t^*}_{\overline{\kappa}(t^*)}$ at stage $t*$. Therefore, $e^t(i, P_i) \neq h^{t^*}_{\overline{\kappa}(t^*)}$ at any $t < t^*$.

left b $h^{t^*}_{\underline{\kappa}(t^*)}$ right b $h^{t^*}_{\overline{\kappa}(t^*)}$

Since $e^t(i, P_i) \neq h^{t^*}_{\overline{\kappa}(t^*)}$ at any $t < t^*$. , and $e^{\bar{t}}(i, P_i) = h^{t^*}_{\overline{\kappa}(t^*)}$, it follows that $\bar{t} \geq t^* + 1$.

For the sake of clarity, we use the notation $H^{t^*}_R(P_i)$ to denote the set $H^{t^*}_R$ when the original preference profile is (P_i, P_{-i}). Similarly, we use the notation $H^{t^*}_R(P'_i)$ when the original preference profile is (P'_i, P_{-i}). By the assumption in Equation 1, we have $H^{t^*}_R(P_i) = H^{t^*}_R(P'_i)$. Let $\widehat{H}_R$ be the set of houses (possibly empty) that will be removed after the stage t^* if we perform right crawler to the set of houses $H^{t^*}_R(P_i)$. We claim that $H^{\hat{\bar{t}}}_R(P'_i) = H^{t^*}_R(P_i) \setminus \widehat{H}_R = H^{\bar{t}}_R(P_i)$. First, note that by the definition of C-TTC, the houses in $\widehat{H}_R$ must be assigned before i is the first right pointing agent. This implies that $H^{\hat{\bar{t}}}_R(P'_i) \subseteq H^{t^*}_R(P_i) \setminus \widehat{H}_R$ and $H^{\bar{t}}_R(P_i) \subseteq H^{t^*}_R(P_i) \setminus \widehat{H}_R$. We need to show no house other than the houses in $\widehat{H}_R$ will be assigned before i is the first right pointing agent. Since $e^{t^*}(i, P_i) = e^{t^*}(i, P'_i) = h^{t^*}_{\underline{\kappa}(t^*)}$ and i is the first right pointing agent at stage $\hat{\bar{t}}$ under P'_i, by the definition of C-TTC, no agent other than the owners of the houses in $\widehat{H}_R$ can be assigned a house in $H^{\hat{\bar{t}}}_R(P'_i)$, and hence $H^{\hat{\bar{t}}}_R(P'_i) = H^{t^*}_R(P_i) \setminus \widehat{H}_R$. Let j be the agent with $e^{t^*}(j, P_i) = h^{t^*}_{\overline{\kappa}(t^*)}$ at stage t^*. Since j has formed a cycle by pointing to the left in C-TTC under (P'_i, P_{-i}) at stage t^* and j's preference does not change in the profile (P_i, P_{-i}), j must point to the left at stage t^* at the profile (P_i, P_{-i}). Since i does not form a cycle involving j at stage t^* under P_i, it must be that i points to some house in $H^{t^*}_M$. Therefore, using similar arguments, it follows that $e^t(j, P_i) = h^{t^*}_{\overline{\kappa}(t^*)}$ for all $t < t'$. Therefore, no agent other than the owners of the houses is assigned any house in $H^{\bar{t}}_R(P_i)$ until the stage $\bar{t}$ in the profile (P_i, P_{-i}). Also, by our assumption, $e^{\bar{t}}(i, P_i) = h^{t^*}_{\overline{\kappa}(t^*)}$ and $\bar{t}$ is the stage when i is the first right pointing agent in $H^{\bar{t}}_R$. Therefore, as we have argued, no agent other than the owners of $\widehat{H}_R$ can be assigned a house in at the profile (P_i, P_{-i}) between stage t^* and stage $\bar{t}$. Combining, we conclude that $H^{\bar{t}}_R(P_i) = H^{t^*}_R(P_i) \setminus \widehat{H}_R$. This completes the proof of $H^{\bar{t}}_R(P_i) = H^{\hat{\bar{t}}}_R(P'_i)$. This completes the proof of the claim.

Since $H^{\bar{t}}_R(P_i) = H^{\hat{\bar{t}}}_R(P'_i)$, which in particular means that the set of available houses for i at stage $\bar{t}$ of the profile (P_i, P_{-i}) is the same as that at stage $\widehat{\bar{t}}$ of the profile (P'_i, P_{-i}), and i receives the most preferred house under P_i from $H^{\bar{t}}_R(P_i)$ at the profile (P_i, P_{-i}), we have $\phi_i(P_i, P_{-i}) R_i \phi_i(P'_i, P_{-i})$.

Case 3. $e^{t^*}(i, P_i) \in H_M^{t^*}$.

If agent i did not form a cycle under P_i, then under P_i', they must form a cycle. Then using similar arguments as in Case 2.2, we can show that agent i cannot be better off by misreporting from P_i to P_i'. Suppose that agent i points to house h and forms a cycle under P_i. If $h \in H_M^{t^*}$, then by the C-TTC algorithm, agent i leaves the algorithm with h at stage $t^* + 1$ and using Lemma .1.1, we have $\phi_i(P_i, P_{-i}) R_i \phi_i(P_i', P_{-i})$. If $h = h^{t^*}_{\underline{\kappa}(t^*)}$(or $h^{t^*}_{\bar{\kappa}(t^*)}$), then we know from the nature of the domain that hR_ih' for every $h' \in H_M^{t^*}$. Lemma .1.2 implies that $\phi_i(P_i, P_{-i}) \preceq^* h$ (or $h \preceq^* \phi_i(P_i, P_{-i})$). Now under P_i' if agent i points to some $h' \in H_M^{t^*}$ then $\phi_i(P_i', P_{-i}) \in H_M^{t^*}$ which implies that $\phi_i(P_i, P_{-i}) R_i \phi_i(P_i', P_{-i})$.

Case 4. $e^{t^*}(i, P_i) = h^{t^*}_{\bar{\kappa}(t^*)}$.

This is symmetric to Case 2.

Case 5. $e^{t^*}(i, P_i) \in H_R^{t^*} \setminus h^{t^*}_{\bar{\kappa}(t^*)}$.

This is symmetric to Case 1.

This completes all the possibilities of $e^{t^*}(i, P_i)$ and concludes the strategy-proof part of the theorem.

Pareto optimality : At each Stage s, the agents who can leave the C-TTC algorithm are the first agent from the left in H_L^s who points weakly to their left, the first agent from the right in H_R^s who points weakly to their right, and any set of agents in H_M^s who form a cycle among themselves. Now from Lemma .1.1, we know that each of these agents are leaving with their most preferred house in the remaining set of houses. Thus starting from the first stage, at each stage the agents leaving the C-TTC algorithm are leaving with their best house in the remaining set of houses. This proves that the C-TTC algorithm satisfies Pareto optimality.

Individual rationality : Fix any agent $i \in N$. Let P_i^* denote the preference of i such that $r_1(P_i^*) = e(i)$. We now argue that for any P_{-i}, $\varphi_i(P_i^*, P_{-i}) = e(i)$. Consider any agent with $e(i) \in H_L$. They must point to their own house at each stage. At every Stage s whenever they is not matched, it must be the case that there is some other agent j such that $e^s(j) \prec^* e^s(i)$ and agent j is pointing to the left of $e^s(j)$. Agent j will leave with their best house h such that $h \prec^* e^s(j) \prec^* e^s(i)$ by the definition of the C-TTC algorithm. Moreover the outcome modification under such cases will be such that $e^s(i) = e^{s+1}(i)$. Finally there will be a stage s^* such that agent i will be the first agent from the left who points weakly to their left. At this stage, agent i will leave the market with $e^{s^*}(i) = e(i)$. One can argue using similar logic for the case of an agent with $e(i) \in H_R$. Finally when $e(i) \in H_M$, agent i will point towards himself at the first stage and form an own cycle and get out of the market with $e(i)$. This completes all the cases and proves that whenever an agent's own endowment is their most preferred house, then they will always be matched that house irrespective of the preference of other agents. Now using the fact that $\phi()$ is strategy-proof, we have $\phi_i(P_i, P_{-i}) R_i \phi_i(P_i^*, P_{-i}) = e(i)$ for every P_i, P_{-i}. This completes the proof that C-TTC algorithm is individually rational.

■

.2 Proof of Theorem 5.4.4

We introduce the following notations to ease the presentation of the proof. For $h, h' \in H$ and for $\prec^* \in \mathbb{L}$, we denote by $[h, h']_{\prec^*}$ the set $\{h'' \in H \mid \text{ either } h \prec^* h'' \prec^* h' \text{ or } h' \prec^* h'' \prec^* h\}$. By $\{h \prec^* \cdots\}$ we denote the set $\{h' \mid h \prec^* h'\}$. Similarly, we use notations like $\{\cdots \prec^* h\}$. For a preference P and a set of houses $\hat{H}$, we denote by $P|_{\hat{H}}$ the restriction of P to the set $\hat{H} \subseteq H$, that is, $P|_{\hat{H}}$ is a preference on $\hat{H}$ with the property that for all $h, h' \in \hat{H}$, we have $hP|_{\hat{H}}h'$ if and only if hPh'.

Proof: We first prove a lemma that provides a crucial structure of the multiple single-peaked domain.

Lemma .2.1 *There exist $h_j, h_k, h_l \in H$ with $h_j \prec^* h_k \prec^* h_l$ and $\prec'^* \in \mathbb{L}$ such that $h_k \notin [h_j, h_l]_{\prec'^*}$.*

Proof:[Proof of Lemma .2.1] It follows from the assumption of Lemma .2.1, that $\prec^*$ and $\prec^{*\prime}$ cannot be opposite of each other.[7] Therefore, there must be $h_t, h_{t+1} \in H$ such that $h_{t+1} \prec'^* h_t$. We complete the proof by distinguishing the following two cases.

Case 1: Either $\{h_{t+1} \prec^* \cdots\} \cap \{h_{t+1} \prec'^* \cdots\} \neq \emptyset$ or $\{\cdots \prec^* h_t\} \cap \{\cdots \prec'^* h_t\} \neq \emptyset$.

Suppose $\{h_{t+1} \prec^* \cdots\} \cap \{h_{t+1} \prec'^* \cdots\} \neq \emptyset$. Let $h_s \in \{h_{t+1} \prec^* \cdots\} \cap \{h_{t+1} \prec'^* \cdots\}$. Then, we have $h_t \prec^* h_{t+1} \prec^* h_s$ and $h_{t+1} \notin [h_t, h_s]_{\prec'^*}$, and thus the requirement of Lemma .2.1 is satisfied when $h_j = h_t$, $h_k = h_{t+1}$, and $h_l = h_s$. Similarly, it follows that the Lemma .2.1 is satisfied when $\{\cdots \prec^* h_t\} \cap \{\cdots \prec'^* h_t\} \neq \emptyset$.

Case 2: $\{h_{t+1} \prec^* \cdots\} \cap \{h_{t+1} \prec'^* \cdots\} = \emptyset$ and $\{\cdots \prec^* h_t\} \cap \{\cdots \prec'^* h_t\} = \emptyset$

[7]An ordering $\prec'$ is called the opposite of another ordering $\prec$ if $h \prec h'$ implies $h' \prec' h$ for all $h, h' \in H$.

Because $\{h_{t+1} \prec^* \cdots\} \cap \{h_{t+1} \prec'^* \cdots\} = \emptyset$ and $\{\cdots \prec^* h_t\} \cap \{\cdots \prec'^* h_t\} = \emptyset$, we have $\{h_{t+1} \prec^* \cdots\} = \{\cdots \prec'^* h_{t+1}\}$ and $\{\cdots \prec^* h_t\} = \{h_t \prec'^* \cdots\}$. We claim that there must be houses h_s and h_u with $h_s \prec^* h_u$ and $h_s \prec'^* h_u$ such that either $h_s, h_u \in \{h_{t+1} \prec^* \cdots\}$ or $h_s, h_u \in \{\cdots \prec^* h_t\}$. This is because, to the contrary, if we have $h_s \prec^* h_u$ and $h_u \prec'^* h_s$ for all $h_s, h_u \in \{h_{t+1} \prec^* \cdots\}$ and all $h_s, h_u \in \{\cdots \prec^* h_t\}$, then $\prec^*$ and $\prec'^*$ turn out to be opposite preferences violating our initial assumption.

Assume without loss of generality that $h_s \prec^* h_u$. Consider the houses h_t, h_s, and h_u. Then, by the choice of those houses, $h_t \prec^* h_s \prec^* h_u$ and $h_s \prec'^* h_u \prec'^* h_t$. This in particular means $h_s \notin [h_t, h_u]_{\prec^*}$, and hence the Lemma .2.1 is satisfied for $h_j = h_t$, $h_k = h_s$, and $h_l = h_u$.

■

We are now ready to prove Theorem 5.4.4. We prove it through contradiction. Let $\mathcal{S}$ be a multiple single peaked domain and assume for contradiction that there exists a matching rule $\phi : \mathcal{S}^n \to \mathcal{M}$ satisfying IR, PO, and obvious strategy-proofness.

Theorem 3.1 in [3] states that a matching rule is obviously strategy-proof if and only if some obviously incentive compatible gradual revelation mechanism implements it. A gradual revelation mechanism is an extensive form mechanism with no singleton action choices at any node and no directly consecutive moves by the same agent. Each non-terminal node $z \in Z$ is associated with a set of preference profiles denoted by $\mathcal{R}_N(v) = \prod_{i \in N} \mathcal{R}_i(v)$ such that $\mathcal{R}(v_0) = \mathcal{S}^n$. At each node v with $\gamma(v) = i$, $\{\mathcal{R}_i(v, a) \mid a \in O(v)\}$ partitions $\mathcal{R}_i(v)$ and $\mathcal{R}_j(v) = \mathcal{R}_j(v, a)$ for all $j \neq i \in N$ and all $a \in O(v)$. A strategy s_i for a player $i \in N$ is said to be truthful in a gradual revelation mechanism if $P_i \in \mathcal{R}_i(v_i, s_i(P_i)(v_i))$ for every non-terminal node $v_i \in V_i$ and all $P_i \in \mathcal{S}$. A gradual revelation mechanism is said to be obviously incentive compatible if the truthful strategy s_i is obviously dominant for each agent $i \in N$. An obviously incentive compatible gradual revelation mechanism M implements a matching rule $\phi : \mathcal{S}^n \to \mathcal{M}$ if $\phi(P_N) = M(\prod_{i \in N} s_i(P_i))$ for all P_N.

In what follows, we show that no obviously incentive compatible gradual revelation mechanism can implement ϕ. Let j, k, l satisfy the condition in Lemma .2.1. For $i \in N \setminus \{j, k, l\}$, consider a preference $\hat{P}_i$ such that $r_1(\hat{P}_i) = h_i$. For $i \in \{j, k, l\}$, consider the set of preferences $\mathbb{P}_i = \{P_i \mid r_1(P_i) \in \{h_j, h_k, h_l\} \setminus \{h_i\}\}$. Let $\mathbb{P}_N = \prod_{i \in N} \mathbb{P}_i$, where $\mathbb{P}_i = \{\hat{P}_i\}$ for $i \in N \setminus \{j, k, l\}$.

Claim .2.1 *For every node v in M, $\mathbb{P}_N \subseteq \mathcal{R}_N(v)$ implies that there exists an action $a \in O(v)$ with $\mathbb{P}_N \subseteq \mathcal{R}_N(v,a)$.*

Proof:[Proof of Claim .2.1] Consider any node v such that $\mathbb{P}_N \subseteq \mathcal{R}_N(v)$. If $\gamma(v) \in N \setminus \{j,k,l\}$, then there exists an action $a \in O(v)$ such that $\hat{P}_{\gamma(v)} \in \mathcal{R}_{\gamma(v)}(v,a)$ since we know from the definition of a gradual revelation mechanism that at each node v, the set $\{\mathcal{R}_{\gamma(v)}(v,a) \mid a \in O(v)\}$ partitions $\mathcal{R}_{\gamma(v)}(v)$. From the definition of a gradual revelation mechanism, we know that $\mathcal{R}_i(v) = \mathcal{R}_i(v,a)$ for each $i \neq \gamma(v)$. Therefore $\mathbb{P}_N \subseteq \mathcal{R}_N(v,a)$.

Now, we consider the case where $\gamma(v) \in \{j,k,l\}$. We present the proof for $\gamma(v) = j$, the proofs for the other two cases follow from similar logic. Consider the preference $P_j \in \mathbb{P}_j$ such that $r_1(P_j) = h_k$ and $h_l P_j h_j$. Suppose there exist actions $a, a' \in O(v)$ such that $P_j \in \mathcal{R}_j(v,a)$ and $P'_j \in \mathcal{R}_j(v,a')$ for some $P'_j \in \mathbb{P}_j$. Since $P'_j \in \mathbb{P}_j$, $r_1(P'_j) \in \{h_k, h_l\}$. Suppose $r_1(P'_j) = h_k$. Consider the preference profile $P_{-j} \in \mathbb{P}_{-j}$ such that $P_i = \hat{P}_i$ for all $i \neq k, l$, $r_1(P_k) = h_l$, and $r_1(P_l) = h_k$ with $h_l P_l h_j$[8].

By individual rationality, we have $\phi_i(P_N) = h_i$ for all $i \in N \setminus \{j,k,l\}$. Further, by individual rationality and Pareto optimality, we have $\phi_k(P_N) = h_l$ and $\phi_l(P_N) = h_k$. This implies $\phi_j(P_N) = h_j$. Consider the preference profile $P'_{-j} \in \mathbb{P}_{-j}$ such that $P'_i = \hat{P}_i$ for all $i \neq k, l$, $r_1(P'_k) = h_j$, and $r_1(P'_l) = h_j$. By individual rationality, we have $\phi_i(P'_N) = h_i$ for all $i \in N \setminus \{j,k,l\}$. Further, by Pareto optimality, $\phi_j(P'_N) \neq h_j$. Therefore, $\phi_j(P'_N) P_j \phi_j(P_N)$ implying that the action a under the truthful strategy at preference P_j is not obviously dominant. Using similar logic, we can argue the case when $r_1(P'_j) = h_l$. Since P'_j is an arbitrary preference in $\mathbb{P}_j$, it follows that the set of preferences $\mathbb{P}_j$ cannot be partitioned through the outgoing edges from the node v, that is, $\mathbb{P}_j \subseteq \mathcal{R}_j(v,a)$.

■

By definition, $\mathbb{P}_N \subseteq \mathcal{R}_N(v_0)$. By the Claim .2.1, there exists an action $a \in O(v_0)$ such that $\mathbb{P}_N \subseteq \mathcal{R}_N(v_0, a)$. Continuing in this manner, we have $\mathbb{P}_N \subseteq \mathcal{R}_N(z)$ for some terminal node $z \in Z$. We know that there exists two different preference profiles in $\mathbb{P}_N$ (P_N and P'_N mentioned earlier) such that $\phi(P_N) \neq \phi(P'_N)$. This along with the fact that P_N and $P'_N \in \mathcal{R}_N(z)$ for some $z \in Z$, we have a contradiction to the obvious strategy proof property of ϕ.

■

[8]Lemma .2.1 allows us full freedom to choose any ordering over h_k, h_l, h_j.

6

Matching under tree single-peaked preferences

6.1 Introduction

6.1.1 Background and description of the problem

We study the classical Shapley-Scarf housing market, consisting of a finite set of agents, each initially endowed with a distinct indivisible house. Each agent holds strict preferences over the entire set of houses. A *matching rule* reallocates houses among the agents based on their reported preferences. Seminal works by [28], [25], and [24] establish that Gale's *Top Trading Cycles* (TTC) algorithm satisfies three fundamental properties: *individual rationality* (IR), *Pareto optimality* (PO), and *strategy-proofness* (SP). Furthermore, [15] proves that when agents' preferences are unrestricted, TTC is the *unique* matching rule satisfying all three properties.

However, this uniqueness result does not extend to *restricted* preference domains. For example, [2] introduces the *crawler* mechanism for the domain of single-peaked preferences and shows that it satisfies IR, PO, and SP. Notably, the crawler also satisfies a stricter incentive requirement, *obvious* *strategy-proofness* (osp), introduced by [13]. This refinement has since motivated a growing body of research into mechanisms that satisfy IR, PO, and osp over structured preference domains (e.g., [30]).

In this paper, we examine the domain of *tree single-peaked preferences*, introduced by [7], which generalizes the classical single-peaked preferences on a line ([2]). Here, the set of alternatives (houses) is connected via a tree graph. A preference is said to be *tree single-peaked* if it declines from the agent's most-preferred house along any path in the tree. Such preferences naturally arise in environments where alternatives are structured via a road or network system without cycles, or when only partial (rather than total) orderings over alternatives are known.

6.1.2 Contribution of the paper

We propose a new algorithm for this domain, called the *Tree Neighborhood Algorithm* (TN algorithm). The TN algorithm assumes a fixed linear order over the set of houses and proceeds in multiple stages. At each stage, a sequence of agents is generated, beginning with the agent who currently holds the highest-ranked house (among unmatched houses). Each agent in the sequence selects their most-preferred house from a permissible subset and points to its current owner, who becomes the next agent in the sequence. This process continues until a cycle is formed. Upon detecting a cycle, all agents in the cycle exchange houses according to the pointing structure. Agents who receive their most-preferred house (among the remaining ones) exit the algorithm with that house; the rest continue into the next stage. The algorithm terminates when all agents have exited.

We show that the TN algorithm satisfies the three desirable properties: *individual rationality*, *Pareto optimality*, and *strategy-proofness*, across all tree single-peaked domains. We also compare the TN algorithm with the crawler algorithm and establish that the crawler is a special case of TN. Moreover, through illustrative examples, we demonstrate that the TN algorithm is *not* outcome-equivalent to the neighborhood TTC algorithm introduced in [14], or the r-neighborhood mechanisms introduced in [9], thereby highlighting the distinctiveness and broader applicability of TN algorithm.

We further extend our analysis to *obvious strategy-proofness* (osp), introduced by [13], a refinement of strategy-proofness that captures limited rationality. A mechanism is osp if, at every decision point in its extensive-form representation, truthful behavior is optimal in a way that is directly verifiable, without requiring contingent reasoning.

In this setting, we establish the following impossibility result: no matching rule satisfies IR, PO, and osp unless the underlying tree is a line graph. This demonstrates that the strong incentive guarantees of the crawler mechanism do not extend to general tree domains.

When the underlying tree is a line (i.e., the classical single-peaked domain), the TN algorithm satisfies osp and strictly generalizes the crawler mechanism. Thus, even within this restricted domain, our analysis identifies a broader class of mechanisms satisfying IR, PO, and osp. Overall, our results complement and extend [2] by providing a sharper structural characterization of incentive-compatible matching mechanisms across both restricted and general preference domains.

6.1.3 Significance of the Paper

Ours is the first paper to provide a class of IR,PO, and SP algorithms on tree single-peaked domains. Single-peaked preferences on trees provide a natural framework for settings where alternatives are embedded in branching network structures. Such environments are common in practice. For instance, residential locations in urban or regional planning are organized along road networks with main routes and branching streets, where agents' preferences decline with distance from an ideal location. School choice systems with feeder patterns also form tree structures, with preferences driven by proximity and accessibility.

Similar structures arise in infrastructure networks such as electricity grids, water and gas distribution systems, where nodes are connected in a tree and preferences depend on network distance. Healthcare referral networks in rural areas and organizational hierarchies within firms likewise exhibit tree structures, inducing single-peaked preferences when agents have an ideal point. More generally, tree-like structures appear in transportation networks, telecommunication systems, river and irrigation networks. In all these cases, alternatives are connected without cycles, and preferences are shaped by proximity to a most-preferred point, underscoring the practical relevance of single-peaked domains on trees.

Our results advance the study of mechanisms satisfying IR, PO, and osp on single-peaked domains. In particular, we identify a significantly larger class of such rules on the classical single-peaked domain, whereas prior to our work, the crawler mechanism was the only known example satisfying all three properties. Moreover, as demonstrated in Sections 6.5.1 and 6.5.1, there exist instances in which the TN algorithm outperforms the crawler in terms of core selection and welfare maximization. This enlarges the feasible set of mechanisms available to a designer aiming to maximize social welfare while maintaining IR, PO, and osp. Our findings also constitute a step toward the broader goal of characterizing all IR, PO, and osp mechanisms on single-peaked domains, which remains an important open problem.

Finally, we complete the investigation of osp mechanisms on tree single-peaked domains by establishing a negative result: no mechanism satisfies IR, PO, and osp unless the underlying tree is a line.

6.1.4 Organization of the Paper

The rest of the paper is organized as follows. Section 6.2 introduces the basic model. In section 6.3, we define and explain the *Tree Neighborhood algorithm* (TN Algorithm). We then state our first theorem that establishes that the TN algorithm satisfies individual rationality, Pareto optimality and strategy-proofness. In section 6.4, we define the notion of obvious strategy-proofness and state the next two results. Finally in section 6.5, we argue that the crawler is a special case of the TN algorithm under single-peaked preferences (with respect to a line graph). All proofs are collected in the appendix.

6.2 Preliminaries

Let $N = \{1, \ldots, n\}$ be a set of agents and let $H = \{h_1, \ldots, h_n\}$ be a set of houses. A (strict) preference over a set S is a binary relation that is complete, transitive, and antisymmetric. Each agent $i \in N$ has a strict preference P_i over the set of houses H. For any preference P, its weak counterpart is denoted by R, where for any $a, b \in H$, aRb if and only if aPb or $a = b$. Given a preference P, the k-th ranked house is denoted by $r_k(P)$, where $r_k(P) = a$ if $|h \in H : hRa| = k$. For a preference P and a subset of houses $\bar{H} \subseteq H$, let $\tau(P, \bar{H})$ denote the most preferred house in $\bar{H}$ according to P, that is, $\tau(P, \bar{H}) \in \bar{H}$ and $\tau(P, \bar{H})Rh$ for all $h \in \bar{H}$. A preference profile is a collection of preferences for all agents and is written as $P_N = (P_1, \ldots, P_n)$. The set of all admissible preferences is denoted by $\mathcal{D}$, and the set of all admissible preference profiles by $\mathcal{D}^n$.

We introduce a particular class of preferences known in the literature as *single-peaked* preferences. A prior order $\prec$ on the set H is a strict ordering of the elements of H.[1] A preference P on H is said to be single-peaked with respect to the prior order $\prec$ if starting from the top-ranked house, preference under P declines as one moves farther away from the peak according to the prior order $\prec$.

Definition 6.2.1 *A preference P on H satisfies single-peakedness with respect to a prior order $\prec$ on H if for all $h, h' \in H$, we have hPh' whenever either $r_1(P) \prec h \prec h'$ or $h' \prec h \prec r_1(P)$.*

We now introduce tree single-peaked preferences. To describe their structure, we first define trees. Let $G = \langle H, E \rangle$ be an undirected graph, where H is the set of nodes (houses) and E is the set of edges. A path from a node h to a node h' is a sequence of distinct nodes $(h^1, h^2, \ldots, h^k)$ such that $(h^i, h^{i+1}) \in E$ for all $i = 1, 2, \ldots, k-1$, with $h^1 = h$ and $h^k = h'$. An undirected graph is said to be connected if there exists a path between every two distinct nodes $h, h' \in H$. Throughout, we restrict attention to connected graphs and refer to them simply as graphs. For any pair of houses $h, h' \in H$, we say that h is adjacent to h' if $(h, h') \in E$. A tree $T = \langle H, E \rangle$ is a graph such that there exists a unique path from any node h to any node h'. Such a unique path in a tree T is denoted by $[h, h']$. By abuse of notation, $[h, h']$ will also denote the set, rather than the sequence, of houses lying on the path from h to h', including both endpoints; the intended meaning will be clear from the context. We follow the convention that parentheses are used to denote an edge and square brackets are used to denote a path. In particular, for any two nodes v and v' in a tree, the expression (v, v') refers to the edge between v and v' whenever such an edge exists. The expression $[v, v']$ refers to the unique path connecting v and v' in the tree.

Definition 6.2.2 *A preferences P on T is said to be* tree single-peaked *with respect to a tree T if for all $h, h' \in H$, we have hRh' whenever h lies on the path from $r_1(P)$ to h', that is, whenever $h \in [r_1(P), h']$.*

We call a set of tree single-peaked preferences $\mathcal{S}(T)$ on a tree T *minimally rich* if, for every $h \in H$ and for any two nodes $h', h'' \in H$ that are adjacent to h, there exist two preferences $P, P' \in \mathcal{S}(T)$ with $r_1(P) = r_1(P') = h$ such that $h'Ph''$ and $h''P'h'$.

[1]Technically, a prior order is a preference. We do not refer to it as a preference since it does not represent an agent's choice behavior.

6.2.1 Matching Rules and Their Properties

A bijection $\mu\colon N \to H$ is called a matching. Let $\mathcal{M}$ denote the set of all matchings. Houses are assigned to agents according to an initial matching ω, which is referred to as the initial endowment. For ease of exposition, whenever the endowment is clear from the context, we use the term 'owner of a house' and the notation $o(h)$ to denote the agent i who is endowed the house h according to the endowment. For instance, when it is clear from the context that we are dealing with the initial endowment, we write $o(h) = i$ if $\omega(i) = h$. Thus, o is a generic notation for the inverse function of the endowment. Without loss of generality, we assume that $\omega(i) = h_i$ for all $i \in N$. A matching rule on a tree single peaked domain is a mapping $\phi\colon \mathcal{S}(T)^n \to \mathcal{M}$ which assigns a matching to each preference profile in $\mathcal{S}(T)^n$.

Next, we define a few desirable properties of a matching rule that are of interest in this paper. A matching μ is said to be individually rational if, for each agent i, the house $\mu(i)$ allocated to i is at least as good as the house $\omega(i)$ with which the agent is endowed. A matching rule is individually rational if it assigns an individually rational matching at every preference profile.

Definition 6.2.3 *A matching $\mu \in \mathcal{M}$ is* ***individually rational (IR)*** *at a preference profile $P_N \in \mathcal{S}(T)^n$ if, for each $i \in N$, we have $\mu(i)R_i\omega(i)$. A matching rule $\phi\colon \mathcal{S}(T)^n \to \mathcal{M}$ is* ***individually rational (IR)*** *if $\phi(P_N)$ is IR at P_N for every $P_N \in \mathcal{S}(T)^n$.*

A matching is Pareto optimal at a given preference profile if there is no other matching such that every agent finds it at least as good as the current matching, and at least one agent finds it strictly better. A matching rule ϕ is Pareto optimal if, for every preference profile R, the matching $\phi(R)$ is Pareto optimal at R.

Definition 6.2.4 *A matching $\mu \in \mathcal{M}$ is* ***Pareto optimal (PO)*** *at a preference profile $P_N \in \mathcal{S}(T)^n$ if there exists no other matching $\mu' \in \mathcal{M}$ such that*

(i) for each $i \in N$, $\mu'(i)R_i\mu(i)$, and

(ii) for some $j \in N$, $\mu'(j)P_j\mu(j)$.

A matching rule $\phi\colon \mathcal{S}(T)^n \to \mathcal{M}$ is ***Pareto optimal (PO)*** *if $\phi(P_N)$ is PO for every $P_N \in \mathcal{S}(T)^n$.*

A matching rule is strategy proof if at every preference profile no agent can be better off by misreporting their true preference.

Definition 6.2.5 *A matching rule $\phi\colon \mathcal{S}(T)^n \to \mathcal{M}$ is* ***strategy proof*** *if for all $i \in N$, all $P_i, P_i' \in \mathcal{S}(T)$, and all $P_{-i} \in \mathcal{S}(T)^{n-1}$, we have*

$$\phi_i(P_i, P_{-i})\, R_i\, \phi_i(P_i', P_{-i}).$$

6.3 Tree Neighborhood Algorithm (TN Algorithm)

In this section, we define the tree neighborhood algorithm. To begin, we introduce several preliminary notions. We first define the concept of a neighbor in this setting.

Let $\bar{H} \subseteq H$, and let $h, h' \in \bar{H}$. We say that h' is a neighbor of h in $\bar{H}$ if the unique path between h and h' intersects $\bar{H}$ only at h and h'. The set of all neighbors of a house h in $\bar{H}$ is denoted by $\mathrm{Nbr}(h, \bar{H})$. We adopt the convention that a house is not regarded as its own neighbor. In particular, for every $h \in \bar{H}$, we have $h \notin \mathrm{Nbr}(h, \bar{H})$.

Similarly, for $\widehat{H} \subseteq H$, a house $h \in \bar{H}$ is said to be a neighbor of the set $\widehat{H}$ in $\bar{H}$ if it is a neighbor of some house in $\widehat{H}$ within $\bar{H}$. That is, there exists a house $h' \in \widehat{H}$ such that $h \in \mathrm{Nbr}(h', \bar{H})$. The set of all neighbors of $\widehat{H}$ in $\bar{H}$ is denoted by $\mathrm{Nbr}(\widehat{H}, \bar{H})$. It is important to note that it is possible that $\widehat{H} \subseteq \mathrm{Nbr}(\widehat{H}, \bar{H})$.

The algorithm is parameterized by an arbitrary but fixed ordering $\prec$ over the houses. The algorithm proceeds in several stages. Each Stage k begins with a set of remaining houses H_k, a set of remaining agents N_k, and an endowment $e_k\colon N_k \to H_k$. Given e_k, we write $o_k(h)$ to denote the owner of a house h according to e_k. In each stage, at least one pair consisting of an agent and a house is permanently assigned and removed from further consideration. Consequently, the algorithm terminates after a finite number of stages.

We present a generic stage, denoted by Stage k, and a generic step within this stage, denoted by Step s. For Stage 1, we set $H_1 = H$, $N_1 = N$, and $e_1 = \omega$.

Stage k

Within Stage k, the steps are defined recursively. For each Step r with $1 \leq r < s$ in Stage k, a house in H_k is selected as a pointer, denoted by $\pi(k, r)$. For every such r, let

$$\Pi(k, r) = \{\pi(k, 1), \ldots, \pi(k, r)\}$$

be the collection of houses selected up to Step r in Stage k.

We now define Step s of Stage k.

- Set $\pi(k, 1) = \tau(\prec, H_k)$. For $s > 1$, define

  $$\pi(k, s) = \tau\big(P_{o_k(\pi(k,s-1))}, \mathrm{Nbr}(\Pi(k, s-1), H_k)\big).$$

  Thus, at each step, the next pointer is determined by the preferences of the owner of the previously selected house, restricted to the neighbors of the houses already chosen in the current stage.

- If

  $$\tau\big(P_{o_k(\pi(k,s))}, H_k\big) \in \Pi(k, s),$$

  say $\tau(P_{o_k(\pi(k,s))}, H_k) = \pi(k, t)$ for some $t \leq s$, then proceed as follows.

  - Define a new endowment e_{k+1} by reassigning, for each $u = t, \ldots, s$, the agent $o_k(\pi(k, u))$ to the house $\pi(k, u+1)$, where we set $\pi(k, s+1) = \pi(k, t)$. For all remaining agents, the endowment remains the same as under e_k.
  - Remove all agent house pairs (i, h) among the houses in $\Pi(k, s)$ such that $e_{k+1}(i) = h$ and h is the most preferred house of agent i among the available houses H_k at Stage k.[2]
  - Proceed to Stage $k+1$, where H_{k+1} consists of the remaining houses, N_{k+1} consists of the remaining agents, and e_{k+1} serves as the initial endowment.

[2] The outcome remains unchanged even if only the agent $o_k(\pi(k, s))$ and the house $\pi(k, s)$ are removed.

- Otherwise, if

$$\tau\big(P_{o_k(\pi(k,s))}, H_k\big) \notin \Pi(k,s),$$

then continue to Step $s+1$ within Stage k.

Since the cardinality of $\Pi(k,s)$ strictly increases at each successive step, there must exist some Step u such that

$$\tau\big(P_{o_k(\pi(k,u))}, H_k\big) \in \Pi(k,u),$$

which guarantees that the sequence of steps in Stage k terminates.

Remark 6.3.1 *Observe that at each step s of every stage k of the algorithm, the owner of the pointer house $o_k(\pi(k,s))$ can choose any house from the set $Nbr(\Pi(k,s), H_k)$. If their most preferred house in H_k lies in $\Pi(k,s)$, then they select that house, implying that a cycle has formed. The agents in the cycle then exchange their current endowments, and stage k of the algorithm terminates. Otherwise, the algorithm proceeds to the next step of stage k. Thus, the set $Nbr(\Pi(k,s), H_k)$ constitutes the "permissible subset" of houses from which the agent selects their most preferred house.*

We illustrate the workings of the TN algorithm with the help of the following example.

Example 6.3.2 *Let $N = \{1, 2, \ldots, 9\}$, where each agent $i \in N$ initially owns house h_i. The table below lists each agent's preferences over the houses, reported up to (and including) their own initial endowment.*

P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9
h_5	h_7	h_1	h_8	h_6	h_7	h_2	h_1	h_1
h_4	h_8	h_2	h_7	h_4	h_9	h_1	h_2	h_2
h_6	h_9	h_3	h_9	h_3	h_4	h_3	h_3	h_3
h_7	h_4	.	h_4	h_7	h_5	h_4	h_4	h_4
h_8	h_3	.	.	h_9	h_6	h_5	h_7	h_7
h_3	h_5	.	.	h_5	.	h_7	h_8	h_9
h_2	h_2	.	.	.	.	.	.	.
h_1	.	.	.	.	.	.	.	.

Table 6.3.1: Preferences of agents

The following diagram depicts the tree structure existing among the houses. Houses are denoted by the nodes and the current owner of a house is displayed above/near the corresponding node.

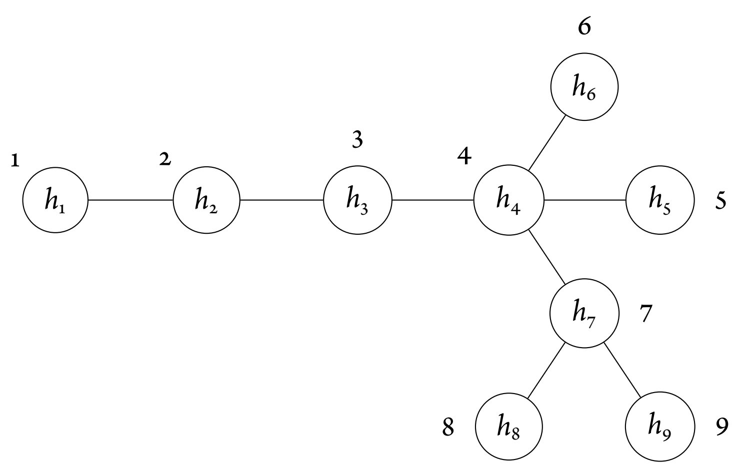

Figure 6.3.1: Initial endowment

Consider the ordering $\prec$ over the houses given by: $h_3 \prec h_2 \prec h_5 \prec h_4 \prec h_1 \prec h_6 \prec h_9 \prec h_7 \prec h_8$. In what follows, we present the TN algorithm with respect to $\prec$.

Stage 1. *At Stage 1, set $H_1 = H$ and $N_1 = N$. The first pointer $\pi(1,1)$ is $\tau(\prec, H_1)$, that is, h_3. Therefore $\Pi(1,1) = \{h_3\}$. Since $\tau(P_3, H_1) = h_1$, which is not in $\Pi(1,1)$, the next pointer $\pi(1,2)$ is selected as the owner* *of the house $\tau(P_3, Nbr(\Pi(1,1), H_1))$. Now $Nbr(\Pi(1,1), H_1) = \{h_2, h_3, h_4\}$. Because $h_2 P_3 h_3 P_3 h_4$, we have $\pi(1,2) = h_2$ and $\Pi(1,2) = \{h_3, h_2\}$.*

Since $\tau(P_2, H_1) = h_7 \notin \Pi(1,2)$, the next pointer $\pi(1,3)$ is selected as the owner of the house $\tau(P_2, Nbr(\Pi(1,2), H_1))$. Since $Nbr(\Pi(1,2), H_1) = \{h_1, h_2, h_3, h_4\}$, we have $\pi(1,3) = h_4$. Similarly, we deduce that $\pi(1,4) = h_7$.

Now since $\tau(P_7, H_1) = h_2$ belongs in $\Pi(1,4)$, we end Stage 1 at this point and update the endowments as follows. To decide the updation, we first note that $\tau(P_7, H_1) = h_2 = \pi(1,2)$. Therefore, the endowment at Stage 2, denoted e_2, is obtained by reallocating agents 2, 4, and 7 to houses h_4, h_7, and h_2, respectively. Since agent 7 receives their most preferred house within H_1, we remove agent 7 along with house h_2 from all subsequent stages. This completes all steps of Stage 1.

The following table summarizes the sequences of pointers and the resulting cycle formed during Stage 1 of the algorithm.

Table 6.3.2: Stage 1: Pointer Sequence and Cycle Formation

Step s	Pointer $\pi(1,s)$	$\Pi(1,s)$	Pointer $\pi(1,s+1)$
1	$\tau(\prec, H_1) = h_3$	$\{h_3\}$	Owner of $\tau(P_3, \{h_2, h_3, h_4\}) = h_2$
2	h_2	$\{h_3, h_2\}$	Owner of $\tau(P_2, \{h_1, h_2, h_3, h_4\}) = h_4$
3	h_4	$\{h_3, h_2, h_4\}$	Owner of $\tau(P_4, \{h_1, h_2, h_3, h_4, h_5, h_6, h_7\}) = h_7$
4	h_7	$\{h_3, h_2, h_4, h_7\}$	$\tau(P_7, H_2) = h_2 \in \Pi(1,4)$: Cycle found
Cycle: $h_2 \to h_4 \to h_7 \to h_2$			
New Endowment e_2**:** $2 \mapsto h_4,\ 4 \mapsto h_7,\ 7 \mapsto h_2$			
Update: Remove agent 7 with house h_2.			

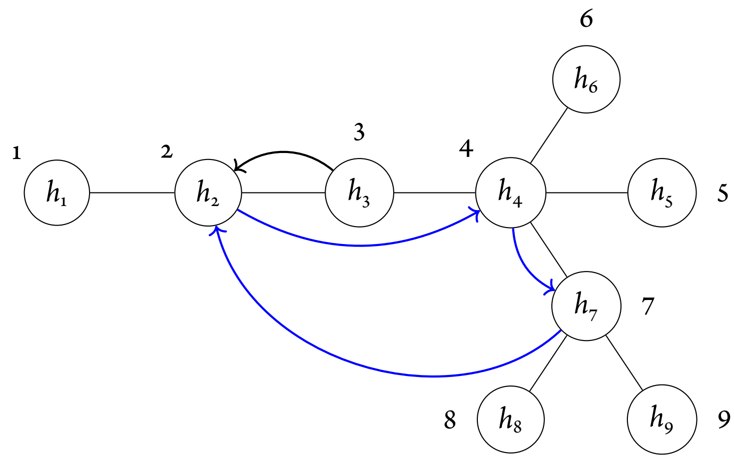

Figure 6.3.2: Stage 1

Figure 6.3.2 summarizes all steps in Stage 1. Here, the arrow from a house points to the pointer for the next step. In Step 1, agent 3 points to house h_2. Therefore, agent 2 starts Step 2. In Step 2, agent 2 points to house h_4, in Step 3, agent 4 points to house h_7, and so on. Whenever an owner of a house points to the house of a previous pointer, we color the edges in the corresponding cycle by blue. The updated endowment for the next stage is decided by moving the owners of the houses according to the blue cycle, and keeping all other endowment unchanged. In subsequent stages, the house-agent pairs who leave the algorithm with their final matching are marked in red.

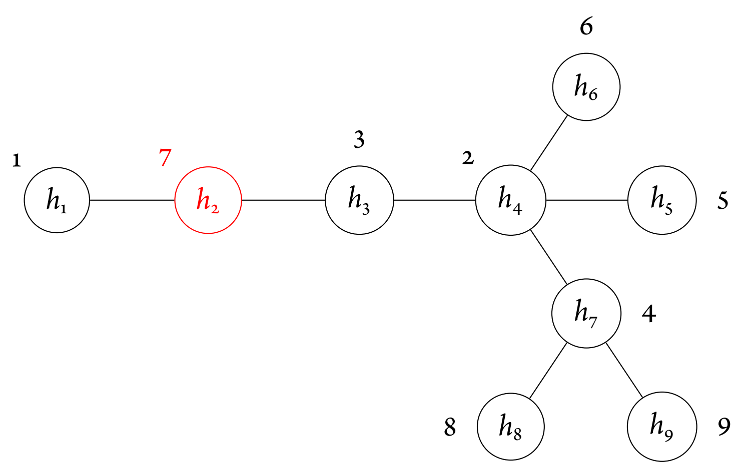

Figure 6.3.3: End of Stage 1

Figure 6.3.3 presents the endowments obtained at the end of Stage 1. The following table summarizes the sequence of pointers and the resulting cycle formed during Stage 2 of the algorithm.

Table 6.3.3: Stage 2: Pointer Sequence and Cycle Formation

Step s	Pointer $\pi(2, s)$	$\Pi(2, s)$	Pointer $\pi(2, s+1)$
1	$\tau(\prec, H_2) = h_3$	$\{h_3\}$	Owner of $\tau(P_3, \{h_1, h_3, h_4\}) = h_1$
2	h_1	$\{h_3, h_1\}$	Owner of $\tau(P_1, \{h_1, h_3, h_4\}) = h_4$
3	h_4	$\{h_3, h_1, h_4\}$	Owner of $\tau(P_2, \{h_1, h_3, h_4, h_5, h_6, h_7\}) = h_7$
4	h_7	$\{h_3, h_1, h_4, h_7\}$	$\tau(P_4, H_2) = h_8$
5	h_8	$\{h_3, h_1, h_4, h_7, h_8\}$	$\tau(P_8, H_2) = h_1 \in \Pi(2, 5)$: Cycle found

Cycle: $h_1 \to h_4 \to h_7 \to h_8 \to h_1$

New Endowment e_3**:** $1 \mapsto h_4,\ 2 \mapsto h_7,\ 4 \mapsto h_8,\ 8 \mapsto h_1$

Update: Remove agents 8, 2, and 4 with houses h_1, h_7, and h_8 respectively.

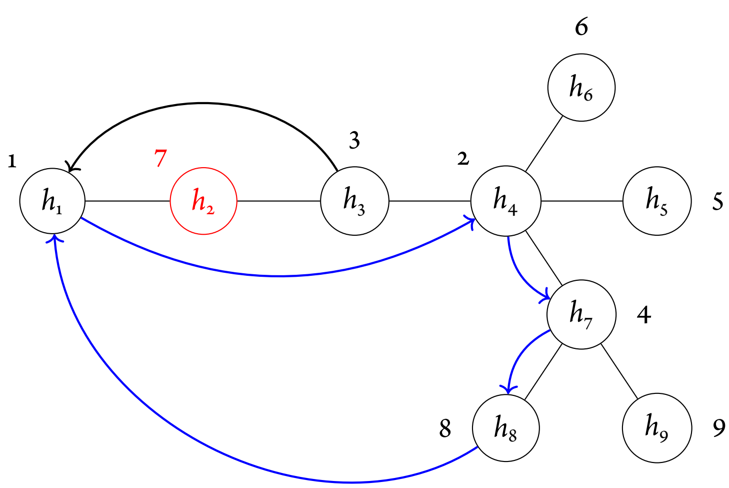

Figure 6.3.4: Stage 2

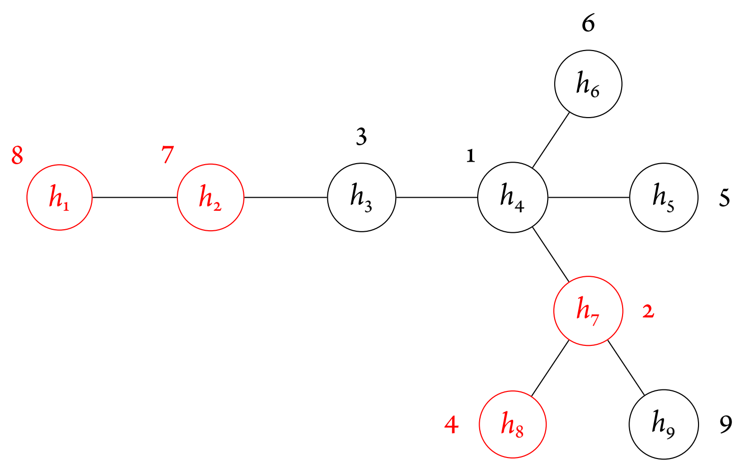

Figure 6.3.5: End of Stage 2

Figure 6.3.4 summarizes all the Steps in Stage 2 and Figure 6.3.5 presents the updated endowment obtained at the end of Stage 2. The subsequent stages of the algorithm proceed analogously. The workings of each stage must be clear to the reader through the illustrative diagram corresponding to that stage.

Table 6.3.4: Stage 3: Pointer Sequence and Cycle Formation

Step s	Pointer $\pi(3, s)$	$\Pi(3, s)$	Pointer $\pi(3, s+1)$
1	$\tau(\prec, H_3) = h_3$	$\{h_3\}$	Owner of $\tau(P_3, \{h_3, h_4\}) = h_3$: Cycle found
Cycle: $h_3 \to h_3$			
New Endowment e_4**:** $3 \mapsto h_3$			
Update: Remove agent 3 with house h_3.			

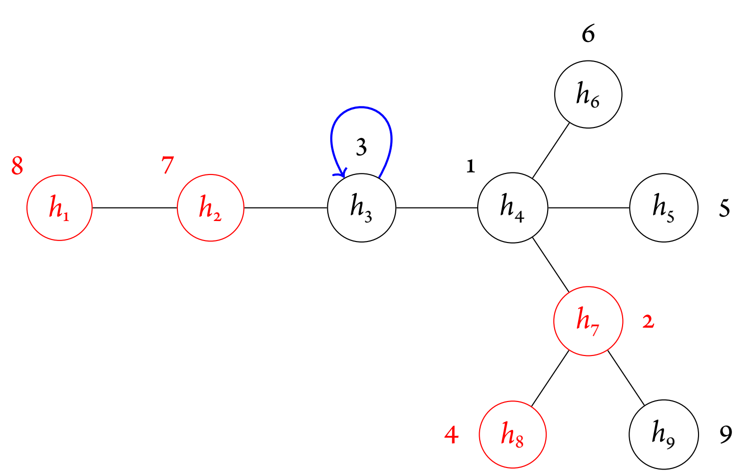

Figure 6.3.6: Stage 3

Table 6.3.5: Stage 4: Pointer Sequence and Cycle Formation

Step s	Pointer $\pi(4,s)$	$\Pi(4,s)$	Pointer $\pi(4,s+1)$
1	$\tau(\prec, H_4) = h_4$	$\{h_4\}$	Owner of $\tau(P_1, \{h_4, h_5, h_6, h_9\}) = h_5$
2	h_5	$\{h_4, h_5\}$	Owner of $\tau(P_5, \{h_4, h_5, h_6, h_9\}) = h_6$
3	h_6	$\{h_4, h_5, h_6\}$	Owner of $\tau(P_6, \{h_4, h_5, h_6, h_9\}) = h_9$
4	h_9	$\{h_4, h_5, h_6, h_9\}$	$\tau(P_9, \{h_4, h_5, h_6, h_9\}) = h_4 \in \Pi(4,4)$: Cycle found
Cycle: $h_4 \to h_5 \to h_6 \to h_9 \to h_4$			
New Endowment e_5**:** $1 \mapsto h_5,\ 5 \mapsto h_6,\ 6 \mapsto h_9,\ 9 \mapsto h_4$			
Update: Remove agents 1, 5, 6, and 8 with houses h_5, h_6, h_9, and h_4, respectively.			

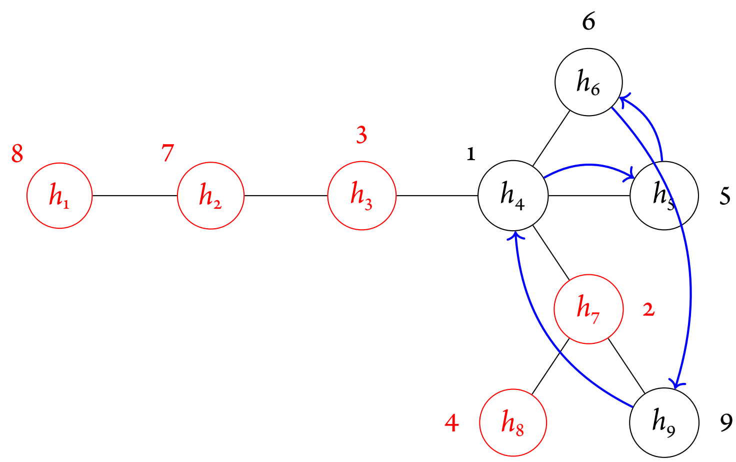

Figure 6.3.7: Stage 4

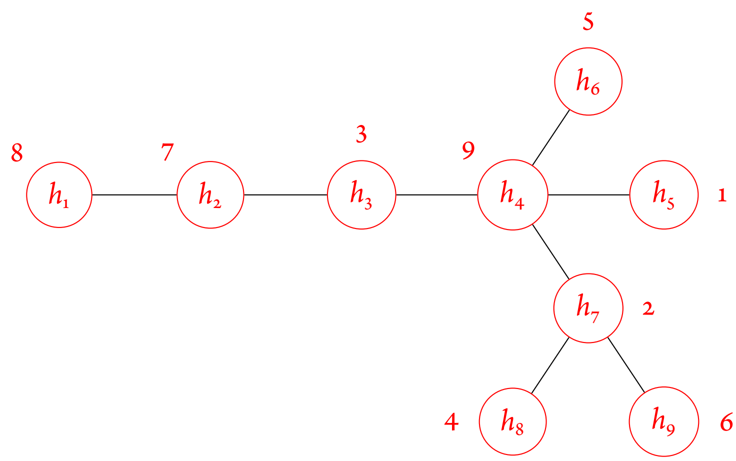

Figure 6.3.8: Final outcome

We now present the main theorem of this section which states that the TN algorithm satisfies the properties of individual rationality, Pareto optimality and strategy-proofness.

Theorem 6.3.3 *The matching rule $\bar{\phi} : \mathcal{S}(T)^n \to \mathcal{M}$ given by the TN algorithm satisfies individual rationality, Pareto optimality and strategy-proofness.*

The proof of this theorem is relegated to Appendix .1.

6.4 Obvious Strategy Proofness

Strategy-proofness ensures that no agent can benefit by misreporting their true preferences. Although it guarantees that truth-telling is a weakly dominant strategy for each agent, agents may not always understand the matching rule well enough to recognize this. [13] introduces the concept of *obvious strategy-proofness*, which identifies rules that are not only strategy-proof but also easily recognizable as such by cognitively limited agents. The paper provides experimental evidence showing that agents behave differently when the mechanism is obviously strategy-proof.

[2] shows that crawler mechanisms satisfy obvious strategy proofness on the single peaked domains defined on a line. However, we establish that when the single peaked domain is generalized to any tree

that is not a line, no such rule exists. This demonstrates that the positive result for single peaked domains on a line is not robust, and it fails even when a single branch is added to the line.

Before proceeding to establish the above mentioned impossibility result, we briefly describe the framework of extensive game forms and provide formal definitions of obvious dominance and obvious strategy proofness.

Definition 6.4.1 *An **extensive game form** is defined as a tuple $\mathbb{E} = \langle \hat{T}, N, \gamma \rangle$, where the components are specified as follows.*

- *The object $\hat{T} = (V, E)$ is a directed rooted tree, where V denotes the set of vertices and E denotes the set of directed edges. The tree has a distinguished root node, denoted by v_0, and a set of terminal nodes denoted by $Z \subseteq V$.*
- *The set N denotes the set of players.*
- *The function $\gamma \colon V \setminus Z \to N$ is the player assigning function. For every non terminal node $v \in V \setminus Z$, the element $\gamma(v)$ specifies the player in N who is assigned to move at node v.*

The set of outgoing edges from a node $v \in V$ is denoted by $O(v)$. For each player $i \in N$, let V_i denote the set of nodes in V at which player i is assigned to move, that is, $V_i = \{ v \in V \mid \gamma(v) = i \}$.

Given an extensive game form $\mathbb{E}$, an action of a player $i \in N$ is defined as a mapping $a_i \colon V_i \to E$ such that, for every node $v_i \in V_i$, we have $a_i(v_i) \in O(v_i)$. Thus, an action of player i specifies one outgoing edge at each node where player i is required to move. We denote by A_i the set of all actions of player i. The set of all action profiles is denoted by $A_N = \prod_{i \in N} A_i$, which consists of one action for each player in N.

A strategy of player $i \in N$ is defined as a mapping $\sigma_i \colon \mathcal{S}(T) \to A_i$. In other words, a strategy of player i assigns to each preference $P_i \in \mathcal{S}(T)$ an action $\sigma_i(P_i) \in A_i$. We denote by Σ_i the set of all strategies of player i. A strategy profile of all players in N is an element $\sigma_N \in \Sigma_N := \prod_{i \in N} \Sigma_i$. For a preference profile $P_N = (P_i)_{i \in N}$, we use the notation $\sigma_N(P_N)$ to denote the action profile $\prod_{i \in N} \sigma_i(P_i)$.

A mechanism is a mapping $m \colon A_N \to \mathcal{M}$. Since each action profile uniquely determines a terminal node of the underlying tree, a mechanism may equivalently be viewed as a function that assigns a matching to every terminal node. We say that a mechanism $m \colon A_N \to \mathcal{M}$ implements a matching rule $\phi \colon \mathcal{S}(T)^n \to \mathcal{M}$ if $\phi(P_N) = m(\sigma_N(P_N))$ for all $P_N \in \mathcal{S}(T)^n$.

Obvious strategy-proofness requires that the mechanism implementing the matching rule admit strategies that are obviously dominant. To determine whether a strategy is obviously dominant at a given preference, fix a player $i \in N$ and consider any node $v \in V_i$ at which player i is required to take an action.

Suppose that, at node v, the strategy prescribes an action $a \in O(v)$. Player i then considers all possible outcomes that may arise from choosing action a at node v. These outcomes correspond to all terminal nodes whose associated paths pass through the edge a. Among these outcomes, player i identifies the worst outcome according to their preference. Next, player i considers any alternative action $a' \in O(v)$ with $a' \neq a$. For each such alternative, player i examines all terminal nodes whose paths pass through the edge a' and determines the best outcome that could result from deviating to that action.

The strategy is said to be obviously dominant if, at every node $v \in V_i$, the worst outcome that can arise from following the prescribed action is weakly preferred by player i to the best outcome that can arise from deviating to any other available action at that node.

For each node $v \in V$, we denote by $\hat{T}^v$ the restriction of the tree $\hat{T}$ to the subtree rooted at v. Let $\mathbb{E}^v$ denote the restriction of the extensive game form $\mathbb{E}$ to this subtree. We refer to $\mathbb{E}^v$ as the subgame extensive form structure rooted at v. Similarly, for a strategy $\sigma_i \in \Sigma_i$ and a strategy profile $\sigma_N \in \Sigma_N$, we denote by σ_i^v and σ_N^v their respective restrictions to the subtree rooted at v.

Definition 6.4.2 *A strategy $\sigma_i \in \Sigma_i$ of player $i \in N$ is said to be obviously dominant in a mechanism m if, for every node $v_i \in V_i$, every preference profile $P_N \in \mathcal{S}(T)^n$, every alternative strategy $\bar{\sigma}_i \in \Sigma_i$ such that $\sigma_i(P_i)(v_i) \neq \bar{\sigma}_i(P_i)(v_i)$, and every strategy profile $\sigma_{-i} \in \Sigma_{N\setminus i}$ of the remaining players, we have*

$$m\big(\sigma_N^{v_i}(P_N)\big)\; R_i \; m\big(\bar{\sigma}_i^{v_i}(P_i), \sigma_{-i}^{v_i}(P_{-i})\big).$$

Definition 6.4.3 *A matching rule $\phi \colon \mathcal{S}(T)^n \to \mathcal{M}$ is said to be obviously strategy proof if there exist a mechanism m and a strategy profile $\sigma_N \in \Sigma_N$ such that*

(i) for every agent $i \in N$, the strategy σ_i is obviously dominant in the mechanism m, and

(ii) the mechanism m implements ϕ, that is, $\phi(P_N) = m(\sigma_N(P_N))$ for all $P_N \in \mathcal{S}(T)^n$.

The next theorem states that, under mild restrictions on the structure of the tree, there does not exist any matching rule that simultaneously satisfies individual rationality, Pareto optimality, and obvious strategy proofness.

Theorem 6.4.4 *Let T be such that there is at least one house $h \in H$ with at least three neighbors. Then there does not exist any matching rule $\phi: \mathcal{S}(T)^n \to \mathcal{M}$ that satisfies individual rationality, Pareto optimality, and obvious strategy proofness.*

The proof of Theorem 6.4.4 is provided in Appendix .2.

In view of Theorem 6.4.4, we now restrict attention to single peaked domains defined on line graphs. Let L be a line graph induced by the ordering $h_1 < \cdots < h_n$. The following theorem states that the TN algorithm satisfies obvious strategy proofness, individual rationality, and Pareto optimality on the single peaked domain $\mathcal{S}(L)$ with respect to the line graph L. As we show in the next section, crawler mechanisms arise as special cases of the TN algorithm for suitable choices of parameters. Therefore, Theorem 6.4.5 generalizes the result in [2] for single peaked domains.

Theorem 6.4.5 *Let T be such that no house $h \in H$ has more than two neighbors. In other words, the tree T reduces to a line graph, denoted by L. The matching rule $\bar{\phi}: \mathcal{S}(L)^n \to \mathcal{M}$ induced by the TN algorithm satisfies individual rationality, Pareto optimality, and obvious strategy-proofness.*

The proof of Theorem 6.4.5 is provided in Appendix .3.

6.5 Discussion

In this section, we compare the TN algorithm with several existing rules from the literature. Since all of these rules are defined on the single peaked domain with respect to a line, we restrict our attention throughout this section to the single peaked domain associated with the line L. Throughout this section, we assume that the line graph L is arbitrary but fixed. For ease of writing, whenever we refer to a single peaked domain, we mean the single peaked domain defined on L.

6.5.1 Comparison with the crawler

[2] introduce the crawler algorithm and show that it satisfies individual rationality, Pareto optimality, and obvious strategy proofness on the single peaked preference domain.

The crawler algorithm operates as follows. At each stage of the algorithm, consider the agent i who occupies the lowest indexed house among those agents who prefer a house with a weakly lower index than their current house. The agent i is then assigned to their most preferred house h and exits the algorithm with house h. The owners of all houses that lie between the current house of agent i and the house h, including the house h, are shifted to the house with the next higher index. All other agents remain in their current houses and continue to participate in the subsequent stages of the algorithm. The procedure terminates when every agent has exited the algorithm.[3]

Crawler is a special case of TN Algorithm

The crawler arises as a special case of the TN algorithm when the ordering $\prec$ coincides with the linear order L. To see this, observe that in the TN algorithm, if agent i is the pointer at Step s of Stage k, then the set $\Pi(k, s)$ consists precisely of those houses that remain available at Stage k and lie weakly to the left of the current house of agent i.

If the most preferred house of agent i among the remaining houses belongs to this set, then agent i is assigned that house. Otherwise, the agent who is endowed with the house immediately to the right of the house currently held by agent i becomes the pointer in Step $s + 1$, and the same procedure is repeated. Consequently, at each stage, the first agent from the left who points to a house weakly to the left of their currently endowed house is assigned that house. It is also straightforward to verify that the subsequent reassignment of houses in the TN algorithm coincides with the rearrangement of houses carried out in the crawler mechanism.

TN algorithm is generally different from crawler

The following example demonstrates that the TN algorithm is, in general, distinct from the crawler algorithm, even on the single peaked domain.

Example 6.5.1 *Let $N = \{1, 2, 3, 4\}$, and suppose that the preferences of the agents are given by the following table.*

[3] For a more formal description of the crawler algorithm, see [2].

P_1	P_2	P_3	P_4
h_4	h_4	h_1	h_1
h_3	h_3	h_2	h_2
h_2	h_2	h_3	h_3
h_1	h_1	h_4	h_4

Table 6.5.1: Preferences of agents

The crawler proceeds as follows. In the first stage, agent 3 occupies the lowest indexed house, namely h_3, among those agents who prefer a house with a weakly lower index than their current house. Since the most preferred house of agent 3 in the set of available houses $\{h_1, h_2, h_3, h_4\}$ at this stage is h_1, agent 3 is assigned to h_1 and exits the algorithm. The owner of h_1, namely agent 1, shifts to h_2, and the owner of h_2, namely agent 2, shifts to h_3.

In the second stage, agent 4, who occupies house h_4, is the next agent who prefers to move to a house with a weakly lower index. The most preferred house of agent 4 in the set of available houses $\{h_2, h_3, h_4\}$ at this stage is h_2. Hence, agent 4 is assigned to h_2 and exits the algorithm. The current owners of h_2 and h_3, namely agents 1 and 2, shift to h_3 and h_4, respectively.

In the third stage, agent 2 is assigned to her most preferred house, namely h_4, in the set of available houses $\{h_3, h_4\}$ and exits the algorithm. No shifting of houses occurs at this stage. Finally, in the last stage, agent 1 remains as the only unmatched agent and is assigned to the last remaining house, namely h_3. The algorithm terminates with the matching (h_3, h_4, h_1, h_2).

In contrast, as shown in Appendix .4, the TN algorithm with respect to the ordering $h_3 \prec h_4 \prec h_1 \prec h_2$ yields the outcome (h_4, h_3, h_2, h_1) at this preference profile, which differs from the outcome of the crawler algorithm.

Remark 6.5.2 *In Example 6.5.1, the TTC outcome at P_N is (h_4, h_3, h_2, h_1), which coincides with the outcome of the TN algorithm. From [23], it is known that the TTC outcome is the unique core. Since the crawler outcome at P_N is different than the TTC outcome, Example 6.5.1 provides a preference profile at which the TN algorithm yields an outcome in the core, whereas the crawler's outcome does not.*

Welfare Comparison

In this section, we demonstrate that the TN algorithm can outperform the crawler in terms of social welfare at certain preference profiles. The purpose of this section is to indicate the possibility that a TN algorithm may be socially optimal under particular welfare criteria. The resolution of this question depends on the choice of a specific welfare function as well as a probability distribution over the space of preference profiles. This constitutes a computationally hard problem, which we leave for future research.

We consider the score vector associated with the widely known Borda social choice function. Let $n = |H|$. For any preference P, the score assigned to the k-th ranked alternative is given by $n - k + 1$. For a matching rule ϕ and a preference profile P_N, the score vector $s_N = (s_i)_{i \in N}$ is defined as a vector of integers, where s_i denotes the score of the outcome assigned to agent $i \in N$ under $\phi(P_N)$. The Borda welfare score of a matching rule ϕ at preference profile P_N is defined as the sum of the components of the score vector.

The following examples provide specific preference profiles for which the TN algorithm yields a higher Borda welfare score than the Crawler. It is worth noting that, with five agents, a welfare gain of at least three units is already substantial.

While there may exist preference profiles where the Crawler performs better in terms of welfare, these instances highlight an important direction for future research, namely, the problem of identifying a socially optimal rule within the broader class of TN rules that satisfies IR, PO, and osp.

Example 6.5.3 *Let $N = \{1, 2, 3, 4, 5\}$, and suppose that $h_3 \prec h_4 \prec h_2 \prec h_5 \prec h_1$. Let the preferences of the agents be given by the following table.*

P_1	P_2	P_3	P_4	P_5
h_2	h_5	h_3	h_2	h_2
h_3	h_4	h_2	h_1	h_3
h_1	h_3	h_1	h_3	h_4
h_4	h_2	h_4	h_4	h_5
h_5	h_1	h_5	h_5	h_1

Table 6.5.2: Preferences of agents

The crawler outcome at P_N is $(h_4, h_5, h_3, h_2, h_1)$ and the score vector of the crawler at P_N is $(2, 5, 5, 5, 1)$. This implies that the Borda welfare score of crawler at P_N is 18.

The TN algorithm outcome at P_N is $(h_4, h_5, h_3, h_1, h_2)$ and the score vector of the TN algorithm at P_N is $(2, 5, 5, 4, 5)$. This implies that the Borda welfare score of TN algorithm at P_N is 21. This shows that at P_N, the Borda welfare score TN algorithm is higher than that of crawler.

There are other preference profiles where the Borda welfare score of TN algorithm is higher than crawler for the same set of agents and linear order $\prec$. A few are listed below.

P'_1	P'_2	P'_3	P'_4	P'_5
h_2	h_4	h_2	h_2	h_5
h_3	h_3	h_1	h_3	h_4
h_4	h_2	h_3	h_1	h_3
h_5	h_5	h_4	h_4	h_2
h_1	h_1	h_5	h_5	h_1

$\bar{P}_1$	$\bar{P}_2$	$\bar{P}_3$	$\bar{P}_4$	$\bar{P}_5$
h_5	h_5	h_2	h_2	h_2
h_4	h_4	h_3	h_3	h_1
h_3	h_3	h_1	h_4	h_3
h_2	h_2	h_4	h_5	h_4
h_1	h_1	h_5	h_1	h_5

Table 6.5.3: Preferences profiles P'_N and $\bar{P}_N$

The crawler outcome at P'_N is $(h_1, h_4, h_2, h_3, h_5)$ and the score vector of the crawler at P'_N is $(1, 5, 5, 4, 5)$. This implies that the Borda welfare score of crawler at P'_N is 20. The TN algorithm outcome at P'_N is $(h_3, h_4, h_1, h_2, h_5)$ and the score vector of the TN algorithm at P'_N is $(4, 5, 4, 5, 5)$. This implies that the Borda welfare score of TN algorithm at P'_N is 23.

The crawler outcome at $\bar{P}_N$ is $(h_4, h_5, h_2, h_1, h_3)$ and the score vector of the crawler at $\bar{P}_N$ is $(4, 5, 5, 1, 3)$. This implies that the Borda welfare score of crawler at $\bar{P}_N$ is 18. The TN algorithm outcome at $\bar{P}_N$ is $(h_5, h_4, h_3, h_2, h_1)$ and the score vector of the TN algorithm at $\bar{P}_N$ is $(5, 4, 4, 5, 4)$. This implies that the Borda welfare score of TN algorithm at $\bar{P}_N$ is 22.

6.5.2 Comparison with the TTC

The following example demonstrates that the TN algorithm is, in general, distinct from the TTC, even on the single-peaked domain.

Example 6.5.4 *Let $N = \{1, 2, 3, 4\}$, and consider the following two preference profiles of the agents, denoted by P_N and P'_N, given in the following tables.*

P_1	P_2	P_3	P_4
h_4	h_4	h_4	h_1
h_3	h_3	h_3	h_2
h_2	h_2	h_2	h_3
h_1	h_1	h_1	h_4

P_1	P'_2	P'_3	P_4
h_4	h_1	h_1	h_1
h_3	h_2	h_2	h_2
h_2	h_3	h_3	h_3
h_1	h_4	h_4	h_4

Table 6.5.4: Preference profiles P_N and P'_N

The TTC outcomes at P_N and P'_N are (h_4, h_2, h_3, h_1) and (h_4, h_2, h_3, h_1) respectively. At P_N, h_4 is the most preferred house of agents 1, 2, 3. For the TN algorithm to be outcome equivalent to TTC at P_N, the $\prec$ ordering cannot be $h_4 \prec \cdots$ or $h_3 \prec \cdots$. This is because, if $h_3 \prec \cdots$, then at Stage 1 of the TN algorithm, $h_3 \to h_4 \to h_2 \to h_4$ and agent 2 will receive h_4 and leave the algorithm. A similar argument can be made for the case when $h_4 \prec \cdots$.

Consider P'_N where h_1 is the most preferred house of agents 2, 3, 4. For the TN algorithm to be outcome equivalent to TTC at P'_N, the $\prec$ ordering cannot be $h_1 \prec \cdots$ or $h_2 \prec \cdots$. This is because, if $h_2 \prec \cdots$, then at Stage 1 of the TN algorithm, $h_2 \to h_1 \to h_3 \to h_1$ and agent 3 will receive h_1 and leave the algorithm. A similar argument can be made for the case when $h_1 \prec \cdots$. Comparing the two sets of constraints on $\prec$ at P_N and P'_N we conclude that there does not exist any $\prec$ such that TN algorithm is outcome equivalent to TTC.

6.5.3 Comparison with the r-neighborhood mechanism

[9] introduced a class of algorithms called the r-neighborhood mechanism and showed that it satisfies individual rationality, Pareto optimality, and group strategy-proofness on the domain of single-peaked preferences. The paper demonstrates that the r-neighborhood mechanism coincides with Gale's Top Trading Cycles (TTC) when $r = n$, and with the crawler mechanism when $r = 1$. Furthermore, it establishes that the r-neighborhood mechanism is obviously strategy-proof only in the extreme case when

$r = 1$.

Below, we provide a brief description of the r-neighborhood mechanism and then show, through an example, that the TN algorithm is not outcome equivalent to the r-neighborhood mechanism.

The r-neighborhood mechanism operates in several stages. For each agent i, currently holding house h_j, fix positive integers l_i^j and r_i^j that define the neighborhood $\mathrm{Nr}(h_j, H_k; l_i^j, r_i^j)$ of agent i with respect to the set of remaining houses H_k at Stage k. The neighborhood set $\mathrm{Nr}(h_j, H_k; l_i^j, r_i^j)$ contains a house h' if and only if $\left|\{\hat{h} \in H_k \mid h' < \hat{h} < h_j\}\right| < l_i^j$ when $h' < h_j$, or $\left|\{\hat{h} \in H_k \mid h_j < \hat{h} < h'\}\right| < r_i^j$ when $h' > h_j$.

At each stage, each agent points to the owner of their most preferred house in $\mathrm{Nr}(h_j, H_k; l_i^j, r_i^j)$ unless they pointed to a non-trading owner in the previous stage, in which case they continue to point to the same owner. Agents in cycles trade, and any agent who is occupying their most preferred house in H_k leaves the mechanism with that house. The process repeats until all agents leave.

Since the r-neighborhood mechanism fails to satisfy obvious strategy-proofness whenever it does not coincide with the crawler (that is, when $r > 1$), whereas the TN algorithm is always obviously strategy-proof, it follows that the TN algorithm differs from the r-neighborhood mechanism. In particular, whenever both mechanisms do not coincide with the crawler, neither is a special case of the other.

To provide further intuition and conceptual clarity, we present the following example to illustrate that the two mechanisms are indeed different.

Example 6.5.5 *Let $N = \{1, 2, 3, 4\}$ and fix the ordering of the TN algorithm as $h_2 \prec h_3 \prec h_4 \prec h_1$. Consider the following two preference profiles of the agents, denoted by P_N and P'_N, given in the tables below.*

P_1	P_2	P_3	P_4
h_4	h_4	h_4	h_1
h_3	h_3	h_3	h_2
h_2	h_2	h_2	h_3
h_1	h_1	h_1	h_4

P_1	P'_2	P'_3	P_4
h_4	h_1	h_1	h_1
h_3	h_2	h_2	h_2
h_2	h_3	h_3	h_3
h_1	h_4	h_4	h_4

Table 6.5.5: Preference profiles P_N and P'_N

As shown in Appendix .5, the TN algorithm yields (h_4, h_2, h_3, h_1) and (h_4, h_2, h_1, h_3) as the outcomes at the *profiles P_N and P'_N, respectively. We prove that no r-neighborhood mechanism can generate these two outcomes at the corresponding profiles.*

At P_N, agent 1 is assigned h_4 and agent 4 is assigned h_1. Since h_4 is the most preferred house of agents 1, 2, and 3, any r-neighborhood mechanism that reproduces this outcome must have the property that the neighborhood ranges of agents 1 and 4 include each other. This, in turn, implies that both agents must have the entire set of houses within their neighborhoods; that is, $r_1^1 = 3$ and $l_4^4 = 3$. However any r-neighborhood mechanism with this neighborhood structure yields the outcome (h_4, h_2, h_3, h_1) at P'_N, which differs from the outcome (h_4, h_2, h_1, h_3) produced by the TN algorithm at P'_N.

We now explain the relative importance of the TN algorithm despite the existence of the *r*-neighborhood mechanism. The *r*-neighborhood mechanism is primarily defined for the single-peaked domain on a line, whereas the TN algorithm applies to a broader class of preference domains. In principle, one could extend the definition of the *r*-neighborhood mechanism to tree single-peaked domains by introducing additional parameters for each house along every path originating from it. However, such an extension would substantially increase the complexity of the mechanism. In contrast, the TN algorithm retains a comparatively simple structure, relying on a single parameter $\prec$.

In the single-peaked domain on a line, the *r*-neighborhood mechanism provides a class of IR, PO, and SP rules that are group strategy-proof. In contrast, the TN algorithm provides a class of IR, PO, and SP rules that satisfy osp. A central goal of this paper is to design mechanisms that satisfy obvious strategy-proofness (osp). For this reason, we do not focus on group strategy-proof mechanisms here.

It is also important to note that group strategy-proofness and osp reflect different design goals. Neither notion is universally more desirable than the other; their relevance depends on the context and the behavioral assumptions being considered. As such, mechanisms satisfying one of these properties should not be viewed as inherently superior to those satisfying the other.

6.5.4 Comparison with the neighborhood TTC

[14] introduce a class of exchange rules called the neighborhood Top Trading Cycles (NTTC) mechanisms and show that these mechanisms satisfy individual rationality, Pareto optimality, and strategy-proofness under single-peaked preferences. The central idea is to restrict, at each stage, the set of

objects available for exchange to a neighborhood, that is, a group of objects that are contiguous in the underlying order that justifies single-peakedness. [14] further show that the NTTC mechanism generalizes Gale's TTC and the crawler for specific structures of *neighborhood trees*.

Below, we provide a brief description of the NTTC mechanisms and then, through an example, show that the TN algorithm is not outcome-equivalent to the NTTC mechanisms.[4] The NTTC mechanism proceeds iteratively, and each iteration consists of two sub-steps:

Preparation Sub-step: Agents whose current endowments are their most preferred objects among the remaining ones are identified and allowed to exit the procedure with their current objects. This step is repeated until no such agent remains.

Exchange Sub-step: Among the remaining agents and objects, a Top Trading Cycles procedure is implemented, but restricted to a subset of objects called the available set, which is determined by a pre-specified rule. Only objects within this set may be exchanged in the current step.

The process then repeats: after each exchange, the preparation sub-step is applied to the updated allocation, followed by another restricted TTC on the corresponding available set. The algorithm terminates once all agents have exited with an object.

The specification of which objects are available for exchange at each step is governed by an availability tree. An availability tree assigns, to every possible partial history of the procedure (i.e., every sequence of past exchanges), a subset of the remaining objects to be made available for the next exchange step. A neighborhood tree is an availability tree in which, at every step, the available set forms a neighborhood, that is, a set of adjacent objects according to the fixed linear order over objects. This ensures that only contiguous groups of objects are eligible for exchange at any given stage. The following example demonstrates that the TN algorithm is different from the NTTC mechanism.

Example 6.5.6 *Let $N = \{1, 2, 3, 4\}$ and let us fix the arbitrary order of the TN algorithm as $h_2 \prec h_4 \prec h_1 \prec h_3$. Now consider the following two preference profiles of the agents P_N and P'_N given by the following tables.*

[4] For a more formal description of the NTTC mechanisms, see [14].

P_1	P_2	P_3	P_4
h_4	h_4	h_4	h_1
h_3	h_3	h_3	h_2
h_2	h_2	h_2	h_3
h_1	h_1	h_1	h_4

P'_1	P_2	P_3	P_4
h_2	h_4	h_4	h_1
h_3	h_3	h_3	h_2
h_4	h_2	h_2	h_3
h_1	h_1	h_1	h_4

Table 6.5.6: Preference profiles P_N and P'_N

The TN algorithm yields the allocations (h_4, h_2, h_3, h_1) *and* (h_2, h_3, h_4, h_1) *at the profiles* P_N *and* P'_N*, respectively.*

First, consider any neighborhood tree whose available set does not include h_4*. At the profile* P_N*, no agent exchanges her current house, since the Top Trading Cycles procedure generates only trivial cycles in which each agent points to her own endowment. Consequently, no agent exits the algorithm, because the most preferred house of agents 1, 2, and 3, namely* h_4*, remains among the unassigned houses.*

Next, suppose that the neighborhood tree includes h_4 *but does not include* h_1*. Let* $h_i \neq h_1$ *be the additional house in the available set together with* h_4*. Then agents i and 4 form a trading cycle and exchange houses, after which agent i exits with her most preferred house,* h_4*. Therefore, the NTTC mechanism can replicate the TN outcome at* P_N *only if the available set of the neighborhood tree contains both* h_1 *and* h_4*, that is, only if it contains the entire set of houses.*

Now consider any neighborhood tree with the above structure at the profile P'_N*. In this case, agents 1, 2, and 4 form a Top Trading Cycle and exit with their respective most preferred houses, namely* h_2*,* h_4*, and* h_1*. Hence, the NTTC outcome at* P'_N *under this neighborhood tree is* (h_2, h_4, h_3, h_1)*, which differs from the TN algorithm outcome* (h_2, h_3, h_4, h_1) *at* P'_N*.*

Appendix

.1 Proof of Theorem 6.3.3

(Strategy-proofness) Consider a preference profile P_N and fix an arbitrary agent i. Suppose that agent i misreports their preference as P'_i.

The outcome of the algorithm depends on the entire preference profile P_N, and therefore, strictly speaking, all induced objects and allocations should be indexed by P_N. However, since only the preference of agent i is altered, we simplify notation by indicating explicitly only the ith component whenever required. Accordingly, we write $e_k(i, P_i)$ instead of $e_k(i, P_N)$ to emphasize the dependence on the reported preference of agent i, while keeping the preferences of all other agents fixed.

We now state and prove several properties of the algorithm that will play a central role in the remainder of the proof.

Claim .1.1 *Given a preference profile P_N, we have $\bar{\phi}_i(P_N)\ R_i\ e_k(i, P_i)$ for all Stages k and for all agents $i \in N$.*

Proof: By definition, $\bar{\phi}_i(P_N)$ is the endowment of agent i at the stage in which i exits the algorithm. Therefore, it suffices to show that $e_{k+1}(i, P_i)\ R_i\ e_k(i, P_i)$ for all $i \in N$ and for every Stage k for which agent i remains active in Stage $k + 1$. Fix an agent $i \in N$ and a Stage k such that i continues in the algorithm at Stage $k + 1$.

If $e_k(i, P_i)$ is not a pointer at any step in Stage k, then $e_{k+1}(i, P_i) = e_k(i, P_i)$, and the claim follows immediately. Suppose instead that $e_{k+1}(i, P_i)$ is a pointer at some Step s in Stage k. If $e_k(i, P_i)$ is not part of the cycle formed at Stage k, then again $e_{k+1}(i, P_i) = e_k(i, P_i)$, and there is nothing to prove. Now suppose that $e_k(i, P_i)$ is part of the cycle formed at Stage k. By the definition of the algorithm, we then have $e_{k+1}(i, P_i) = \tau\big(P_i, \mathrm{Nbr}(\Pi(k, s), H_k)\big)$. Since $e_k(i, P_i) \in \mathrm{Nbr}(\Pi(k, s), H_k)$, it follows from the definition of τ that $e_{k+1}(i, P_i)\ R_i\ e_k(i, P_i)$. This completes the proof. ■

We now introduce certain definitions that will be used in the proofs of the subsequent lemmas.

Let $T = \langle V, E\rangle$ be a tree, and let $\overline{V}$ and $\widehat{V}$ be subsets of V such that $\overline{V} \subseteq \widehat{V}$. We say that $\overline{V}$ is a connected component of T restricted to $\widehat{V}$ if for any $v, v' \in \overline{V}$, the path $[v, v']$ in T satisfies $[v, v'] \cap \big(\widehat{V} \setminus \overline{V}\big) = \emptyset$.

For any subset $\widehat{V} \subseteq V$ and any path $(v_1, \ldots, v_m)$ in T, we define the path $(v_1, \ldots, v_m)$ restricted to $\widehat{V}$ to be the subsequence $(v_{l_1}, \ldots, v_{l_k})$ of $(v_1, \ldots, v_m)$ such that $\widehat{V} \cap \{v_1, \ldots, v_m\} = \{v_{l_1}, \ldots, v_{l_k}\}$, where the order is inherited from the original path.

Lemma .1.1 *Suppose $x = \pi(k, t)$ and $y = \pi(k, \hat{t})$ for some Stage k and some Stes t and $\hat{t}$ in Stage k with $t < \hat{t}$. Then, for all $z \in [x, \ldots, y]$,[5] there exists a Stage j and a Step u in Stage j with $(j, u) < (k, \hat{t})$ such that*

[5] Whenever we refer to a path without further restriction, we mean the path in the original tree T.

$z = \pi(j, u)$.[6]

Proof: Assume, for contradiction, that there exists $z \in [x, \ldots, y]$ that violates the assertion of the lemma. Write $\hat{t} = t + c$ for some positive integer c.

If $z \notin H_k$, then z must have exited the algorithm in some previous Stage j. In that case, there exists a Step u in Stage j such that z is a pointer, that is, $z = \pi(j, u)$. This contradicts the assumption.

Suppose therefore that $z \in H_k$. We complete the proof by showing that $\Pi(k, t + c)$ forms a connected component of T restricted to H_k. Observe that $\Pi(k, 1)$ is a singleton set consisting of $\pi(k, 1)$. Moreover, for all $s \geq 1$, we have $\pi(k, s + 1) \in \mathrm{Nbr}(\Pi(k, s), H_k)$. Since $\pi(k, s + 1)$ is a neighbor of $\Pi(k, s)$ within H_k, it follows that if $\Pi(k, s)$ is a connected component of T restricted to H_k, then the set
$\Pi(k, s + 1) = \Pi(k, s) \cup \{\pi(k, s + 1)\}$ is also a connected component of T restricted to H_k. This completes the proof. ■

Before proceeding with the remaining lemmas required for the proof of Theorem 6.3.3, we define projections and establish an important property of projections in the following claim. Let $\bar{H} \subseteq H$ and let $h \in \mathrm{Nbr}(\bar{H}, H) \setminus \bar{H}$. We say that $h' \in \bar{H}$ is a projection of h on $\bar{H}$ if $h \in \mathrm{Nbr}(h', H)$.

Claim .1.2 *For every $x \in \mathrm{Nbr}(\Pi(k, s), H_k) \setminus \Pi(k, s)$ for some Stage k and some Step s in Stage k, there exists a unique projection of x on $\Pi(k, s)$.*

Proof: Assume, for contradiction, that x' and x'' are two distinct projections of x on $\Pi(k, s)$. By definition, this implies that $x', x'' \in \Pi(k, s)$. Hence there exist Stes t' and t'' in Stage k such that $x' = \pi(k, t')$ and $x'' = \pi(k, t'')$.

By Lemma .1.1, for every node $z \in [x', \ldots, x'']$, there exists a Stage $j \leq k$ and a Step u in Stage j with $(j, u) \leq (k, s)$ such that $z = \pi(j, u)$. Since $x \in \mathrm{Nbr}(\Pi(k, s), H_k) \setminus \Pi(k, s)$ and in particular $x \in H_k$, it follows that $x \notin [x', \ldots, x'']$.

Because x' and x'' are projections of x on $\Pi(k, s)$, the path $[x, \ldots, x']$ that does not contain x'', and the path $[x, \ldots, x'']$ does not contain x'. Consequently, the three paths $[x, \ldots, x']$, $[x', \ldots, x'']$, and $[x, \ldots, x'']$ together induce a cycle in the graph. This contradicts the fact that T is a tree. Therefore the projection of x on $\Pi(k, s)$ is unique. ■

[6] The ordering $(j, u) \leq (k, \hat{t})$ is with respect to the lexicographic order, that is, either $j < k$, or $j = k$ and $u < \hat{t}$.

Lemma .1.2 *Let $x, y \in \mathrm{Nbr}(\Pi(k,s), H_k) \setminus \Pi(k,s)$ for some Stage k and some Step s in Stage k. Suppose x' and y' are the projections of x and y on $\Pi(k,s)$ respectively, that is, $x', y' \in \Pi(k,s)$, $x \in \mathrm{Nbr}(x', H_k)$, and $y \in \mathrm{Nbr}(y', H_k)$. Then the path $[x, \ldots, y]$ is of the form $[x, \ldots, x', \ldots, y', \ldots, y]$.*[7]

Proof: We first note that the set $\Pi(k,s)$ forms a connected component of T restricted to H_k. Since x' and y' belong to $\Pi(k,s)$ and the underlying graph is a tree, the unique path $[x', y']$ from x' to y' in H_k must lie entirely in $\Pi(k,s)$. Hence $[x', \ldots, y'] \subseteq \Pi(k,s)$.

Since x and y do not belong to $\Pi(k,s)$ and the path $[x', \ldots, y']$ lies in $\Pi(k,s)$, it follows that $x, y \notin [x', \ldots, y']$. Moreover, x' and y' are projections of x and y respectively, which means in particular that x and y are neighbors of x' and y' in H_k respectively. Using the definition of neighbors and the fact that $\Pi(k,s) \subseteq H_k$, it follows that $[x, \ldots, x'] \cap \Pi(k,s) = \{x'\}$ and $[y, \ldots, y'] \cap \Pi(k,s) = \{y'\}$.

Therefore, by concatenating the paths $[x, \ldots, x']$, $[x', \ldots, y']$, and $[y', \ldots, y]$, we obtain $[x, \ldots, y] = [x, \ldots, x', \ldots, y', \ldots, y]$. This completes the proof. ■

Lemma .1.3 *Let $i = o(\pi(k,s))$ for some Stage k and some Step s in Stage k. If $\tau(P_i, H_k) \notin \Pi(k,s)$ and $x = \tau(P_i, \mathrm{Nbr}(\Pi(k,s), H_k))$, then $\bar{\phi}_i(P_i, P_{-i})\, R_i\, x'$, where x' is the projection of x on $\Pi(k,s)$.*

Proof: Suppose first that $e_{k+1}(i, P_i) \neq e_k(i, P_i)$. Then, by the definition of the algorithm, $e_{k+1}(i, P_i) = x$. By Claim .1.1, we have $\bar{\phi}_i(P_i, P_{-i})\, R_i\, x$. Since $x = \tau(P_i, \mathrm{Nbr}(\Pi(k,s), H_k))$ and $x' \in \mathrm{Nbr}(\Pi(k,s), H_k)$, it follows that $x\, R_i\, x'$. Therefore $\bar{\phi}_i(P_i, P_{-i})\, R_i\, x'$.

Now suppose that $e_{k+1}(i, P_i) = e_k(i, P_i)$. Let t be the final step of Stage k. Since $\tau(P_i, H_k) \notin \Pi(k,s)$ and $e_{k+1}(i, P_i) = e_k(i, P_i)$, it follows from the definition of the algorithm that the owner of the pointer $\pi(k,t)$ points to some house $\pi(k,u)$ with $u > s$. Hence the houses $\pi(k,1), \pi(k,2), \ldots, \pi(k,s)$ are not involved in the cycle formed in Stage k, and therefore these houses remain in the algorithm for the next stage with their respective owners. Consequently $\Pi(k+1, s') = \Pi(k, s')$ for all $s' = 1, 2, \ldots, s$, and in particular $o(\pi(k+1, s)) = i$.

At the sth Step of Stage $k+1$, either agent i points to some house in $\Pi(k+1, s)$ and receives that house and exits the algorithm, or agent i points to some house in $\mathrm{Nbr}(\Pi(k+1,s), H_{k+1}) \setminus \Pi(k+1,s)$.

[7] We use the notation $[x, \ldots, x', \ldots, y', \ldots, y]$ to emphasize that the unique path from x to y contains the nodes x' and y' such that x' belongs to the unique path from x to y'. We use similar notation without further explanation.

In the former case, $\bar{\phi}_i(P_i, P_{-i})$ is the most preferred house for agent i according to P_i among the set $\{\pi(k+1, 1), \ldots, \pi(k+1, s)\}$. Since this set coincides with $\{\pi(k, 1), \ldots, \pi(k, s)\}$ and x' belongs to this set, it follows that $\bar{\phi}_i(P_i, P_{-i})\, R_i\, x'$. In the latter case, either $e_{k+2}(i, P_i) \neq e_{k+1}(i, P_i)$ or $e_{k+2}(i, P_i) = e_{k+1}(i, P_i)$. If $e_{k+2}(i, P_i) \neq e_{k+1}(i, P_i)$, then by an argument analogous to the first part of the proof, we obtain $\bar{\phi}_i(P_i, P_{-i})\, R_i\, x'$. If $e_{k+2}(i, P_i) = e_{k+1}(i, P_i)$, we repeat the same reasoning for subsequent stages.

Since the algorithm terminates after finitely many stages, there exists a Stage k^* such that either agent i points to a house in $\Pi(k^*, s)$ and receives that house as the final allocation, or agent i points to a house in $\mathrm{Nbr}(\Pi(k^*, s), H_{k^*}) \setminus \Pi(k^*, s)$ and receives that house as the final allocation, in which case $e_{k^*+1}(i, P_i) \neq e_{k^*}(i, P_i)$. In each case, it follows that $\bar{\phi}_i(P_i, P_{-i})\, R_i\, x'$. This completes the proof. ■

We are now in a position to complete the proof of Theorem 6.3.3. Let $\hat{k}+1$ be the first stage such that $e_{\hat{k}+1}(i, P_i) \neq e_{\hat{k}+1}(i, P_i')$. Then, at some Step $\hat{s}$ of Stage $\hat{k}$, the house $e_{\hat{k}}(i, P_i)$ acts as a pointer. Since $e_{\hat{k}}(i, P_i) = e_{\hat{k}}(i, P_i')$ by assumption, and the preferences of agents other than i remain unchanged, the fact that agent i is a pointer at Step $\hat{s}$ under (P_i, P_{-i}) implies that agent i is also a pointer at Step $\hat{s}$ under (P_i', P_{-i}).

If $\tau(P_i, H_{\hat{k}}) \in \Pi(\hat{k}, \hat{s})$, then agent i receives the most preferred available house at Stage $\hat{k}$ and therefore cannot benefit from manipulation. Suppose instead that $\tau(P_i, H_{\hat{k}}) \notin \Pi(\hat{k}, \hat{s})$ and agent i points to the house $x = \tau(P_i, \mathrm{Nbr}(\Pi(\hat{k}, \hat{s}), H_{\hat{k}}))$.

We consider the following possible cases of manipulation:

(I) $\tau(P_i', H_{\hat{k}}) \in \Pi(\hat{k}, \hat{s})$.

(II) $\tau(P_i', H_{\hat{k}}) \notin \Pi(\hat{k}, \hat{s})$ and $\tau(P_i', \mathrm{Nbr}(\Pi(\hat{k}, \hat{s}), H_{\hat{k}})) \neq \tau(P_i, \mathrm{Nbr}(\Pi(\hat{k}, \hat{s}), H_{\hat{k}}))$.

Case (I): Since $\tau(P_i', H_{\hat{k}}) \in \Pi(\hat{k}, \hat{s})$, it follows from the definition of the algorithm that $\bar{\phi}_i(P_i', P_{-i}) = \tau(P_i', H_{\hat{k}})$. By Lemma .1.3, we have $\bar{\phi}_i(P_i, P_{-i})\, R_i\, x'$, where x' is the projection of x on $\Pi(\hat{k}, \hat{s})$.

Since both $\bar{\phi}_i(P_i', P_{-i})$ and x' belong to $\Pi(\hat{k}, \hat{s})$, and since $x \in H_{\hat{k}} \setminus \Pi(\hat{k}, \hat{s})$, Lemma .1.1 implies that there exists a path $[\bar{\phi}_i(P_i', P_{-i}), \ldots, x']$ that does not contain x. Moreover, because

$\tau(P_i, \mathrm{Nbr}(\Pi(\hat{k}, \hat{s}), H_{\hat{k}})) = x$, it follows from the definition of the algorithm that $x \in [x', \ldots, \tau(P_i, H_{\hat{k}})]$. Concatenating these two paths yields a path

$$[\bar{\phi}_i(P'_i, P_{-i}), \ldots, x', \ldots, x, \ldots, \tau(P_i, H_{\hat{k}})].$$

By the single peakedness of the preferences, it follows that $x' \, R_i \, \bar{\phi}_i(P'_i, P_{-i})$. Since $\bar{\phi}_i(P_i, P_{-i}) \, R_i \, x'$, we obtain $\bar{\phi}_i(P_i, P_{-i}) \, R_i \, \bar{\phi}_i(P'_i, P_{-i})$.

Case (II): Suppose that $\tau(P_i, \mathrm{Nbr}(\Pi(\hat{k}, \hat{s}), H_{\hat{k}})) = x$ and $\tau(P'_i, \mathrm{Nbr}(\Pi(\hat{k}, \hat{s}), H_{\hat{k}})) = y$. Let x' and y' be the projections of x and y on $\Pi(\hat{k}, \hat{s})$, respectively.

Under (P_i, P_{-i}) we have $x = \pi(\hat{k}, \hat{s}+1)$. By Lemma .1.3, $\bar{\phi}_i(P_i, P_{-i}) \, R_i \, x'$. From the argument in Case (I), we know that $x \in [x', \ldots, \tau(P_i, H_{\hat{k}})]$. By the assumption on x and y, Lemma .1.2 implies that $[x, \ldots, x', \ldots, y', \ldots, y]$ constitutes a path. Since $x \in [x', \ldots, \tau(P_i, H_{\hat{k}})]$, it follows that

$$[\tau(P_i, H_{\hat{k}}), \ldots, x, \ldots, x', \ldots, y', \ldots, y]$$

constitutes a path. By single peakedness, we obtain $x' \, R_i \, y'$.

Under (P'_i, P_{-i}) we have $y = \pi(\hat{k}, \hat{s}+1)$. Applying analogous arguments with the roles of x and y interchanged, the sequence

$$[\tau(P'_i, H_{\hat{k}}), \ldots, y, \ldots, y', \ldots, x', \ldots, x]$$

constitutes a path. Since the underlying graph is a tree and hence admits a unique path between every two nodes, these two paths together imply that

$$[\tau(P_i, H_{\hat{k}}), \ldots, x, \ldots, x', \ldots, y', \ldots, y, \ldots, \tau(P'_i, H_{\hat{k}})] \tag{1}$$

constitutes a path.

By Lemma .1.3, $\bar{\phi}_i(P'_i, P_{-i}) \, R'_i \, y'$. If $\bar{\phi}_i(P'_i, P_{-i}) \in \{y, y'\}$, then single peakedness of P_i together with the fact that (1) is a path implies that x' is preferred to both y and y' according to P_i. Since $\bar{\phi}_i(P_i, P_{-i}) \, R_i \, x'$, we conclude that $\bar{\phi}_i(P_i, P_{-i}) \, R_i \, \bar{\phi}_i(P'_i, P_{-i})$.

Now suppose that $\bar{\phi}_i(P'_i, P_{-i}) \notin \{y, y'\}$. Let z denote the projection of $\bar{\phi}_i(P'_i, P_{-i})$ on the path given in (1). Then $z \in [y, \ldots, \tau(P'_i, H_{\hat{k}})] \setminus \{y\}$. Because (1) constitutes a path, we have $x' \, R_i \, z$. Moreover, since z is the projection of $\bar{\phi}_i(P'_i, P_{-i})$ on that path, it follows that $z \, R_i \, \bar{\phi}_i(P'_i, P_{-i})$. Combining these relations yields $x' \, R_i \, \bar{\phi}_i(P'_i, P_{-i})$. Using the same argument as before, we conclude that $\bar{\phi}_i(P_i, P_{-i}) \, R_i \, \bar{\phi}_i(P'_i, P_{-i})$.

(Pareto optimality) Pareto optimality of the algorithm follows from the observation that an agent i is allocated a house h at Stage k only if h is the most preferred house of agent i among the available houses H_k at that stage according to P_i. Hence, no agent can be made strictly better off without making some other agent worse off.

(Individual rationality) Claim .1.1 establishes that $\bar{\phi}_i(P_i, P_{-i}) \, R_i \, e_k(i)$ for all Stages k. In particular, this implies that $\bar{\phi}_i(P_i, P_{-i}) \, R_i \, e_1(i)$, which proves individual rationality.

.2 Proof of Theorem 6.4.4

Let $\bar{h}$ be a house in T that has at least three neighbors. Let h_j, h_k, and h_l be three distinct neighbors of $\bar{h}$. Consider the collection of preference orderings $\mathbb{P}$ in which $\bar{h}$ is ranked first and the houses h_j, h_k, and h_l are ranked immediately after $\bar{h}$ in all possible orders. Formally,

$$\mathbb{P} = \{\bar{h}h_jh_kh_l \ldots, \bar{h}h_jh_lh_k \ldots, \bar{h}h_kh_jh_l \ldots, \bar{h}h_kh_lh_j \ldots, \bar{h}h_lh_jh_k \ldots, \bar{h}h_lh_kh_j \ldots\}.$$

Suppose, for the sake of contradiction, that there exists a matching rule $\phi : \mathcal{S}(T)^n \to \mathcal{M}$ that satisfies individual rationality, Pareto optimality, and obvious strategy proofness. By Theorem 3.1 in [3], a matching rule $\phi : \mathcal{S}(T)^n \to \mathcal{M}$ is obviously strategy proof if and only if there exists an obviously incentive compatible gradual revelation mechanism Γ that implements it.

Gradual revelation mechanism : A gradual revelation mechanism is a mechanism whose underlying extensive form game satisfies the following properties.

There are no singleton action choices at any decision node, and no agent moves at two adjacent nodes. Each non terminal node $v \in Z$ is associated with a set of preference profiles denoted by $\mathcal{R}_N(v) = \prod_{i \in N} \mathcal{R}_i(v)$, with the property that $\mathcal{R}_N(v_0) = \mathcal{S}(T)^n$.

For a node v and an outgoing edge a from v, we denote by (v, a) the successor node v' that follows from v through the edge a, that is, the edge from v to v' is a. At any node v such that $\gamma(v) = i$, the collection $\{\mathcal{R}_i(v, a) \mid a \in O(v)\}$ forms a partition of $\mathcal{R}_i(v)$. Moreover, $\mathcal{R}_j(v) = \mathcal{R}_j(v, a)$ for all $j \neq i$ and for all $a \in O(v)$.

A strategy σ_i for agent $i \in N$ is said to be truthful in a gradual revelation mechanism if, for every non terminal node $v_i \in V_i$ and every $P_i \in \mathcal{S}(T)$, one has $P_i \in \mathcal{R}_i\big(v_i, \sigma_i(P_i)(v_i)\big)$. The mechanism is obviously incentive compatible if the truthful strategy σ_i is obviously dominant for every agent $i \in N$.

An obviously incentive compatible gradual revelation mechanism Γ implements the matching rule $\phi : \mathcal{S}(T)^n \to \mathcal{M}$ if $\phi(P_N) = \Gamma\big(\sigma_N(P_N)\big)$ for all $P_N \in \mathcal{S}(T)^n$.

In what follows, we show that no obviously incentive compatible gradual revelation mechanism Γ can implement ϕ. We first construct a subset of the preference domain as follows. For each $i \notin \{j, k, l\}$, fix an arbitrary preference $\hat{P}_i \in \mathcal{S}(T)$ such that $r_1(\hat{P}_i) = h_i$. For each $i \in \{j, k, l\}$, define

$$\mathbb{P}_i = \{P_i \in \mathcal{S}(T) \mid P_i \in \mathbb{P} \text{ and } r_2(P_i) \neq h_i\}.$$

Let $\mathbb{P}_N = \prod_{i \in N} \mathbb{P}_i$, where for every $i \notin \{j, k, l\}$ we set $\mathbb{P}_i = \{\hat{P}_i\}$.

Claim .2.1 *For every node v in Γ, if $\mathbb{P}_N \subseteq \mathcal{R}_N(v)$, then there exists an edge $a \in O(v)$ such that $\mathbb{P}_N \subseteq \mathcal{R}_N(v, a)$.*

Proof:[Proof of Claim .2.1] Fix a node v such that $\mathbb{P}_N \subseteq \mathcal{R}_N(v)$. First suppose that $\gamma(v) \in N \setminus \{j, k, l\}$. By the definition of a gradual revelation mechanism, the collection $\{\mathcal{R}_{\gamma(v)}(v, a) \mid a \in O(v)\}$ partitions $\mathcal{R}_{\gamma(v)}(v)$. Since $\hat{P}_{\gamma(v)} \in \mathcal{R}_{\gamma(v)}(v)$, there exists an edge $a \in O(v)$ such that $\hat{P}_{\gamma(v)} \in \mathcal{R}_{\gamma(v)}(v, a)$. Moreover, for every $i \neq \gamma(v)$, we have $\mathcal{R}_i(v, a) = \mathcal{R}_i(v)$. Hence $\mathbb{P}_N \subseteq \mathcal{R}_N(v, a)$.

Next suppose that $\gamma(v) \in \{j, k, l\}$. We present the argument for the case $\gamma(v) = j$, since the other cases follow by symmetry.

Consider a preference $P_j' \in \mathbb{P}_j$ given by $P_j' = \bar{h}h_kh_lh_j \ldots$. Suppose there exist two distinct edges $a, a' \in O(v)$ and another preference $P_j'' \in \mathbb{P}_j$ such that $P_j' \in \mathcal{R}_j(v, a)$ and $P_j'' \in \mathcal{R}_j(v, a')$. Since $P_j'' \in \mathbb{P}_j$, we have $P_j'' = \bar{h}h_m \ldots$ for some $m \in \{k, l\}$. Let us assume that $P_j'' = \bar{h}h_k \ldots$. The argument for the case $P_j'' = \bar{h}h_l \ldots$ proceeds in an analogous manner.

Now construct a profile $P'_N = (P'_j, P'_{-j})$ as follows. For every $i \notin \{j, k, l\}$, set $P'_i = \hat{P}_i$. Let $P'_k = \bar{h}h_l h_k h_j \dots$ and $P'_l = \bar{h}h_k h_l h_j \dots$. Since ϕ satisfies individual rationality, we have $\phi_i(P'_N) = h_i$ for all $i \notin \{j, k, l\}$. Hence the houses $\{h_j, h_k, h_l\}$ are assigned among the agents $\{j, k, l\}$. By individual rationality and Pareto optimality, it follows that $\phi_k(P'_N) = h_l$ and $\phi_l(P'_N) = h_k$. Therefore $\phi_j(P'_N) = h_j$.

Next consider the profile $P''_N = (P''_j, P''_{-j})$ constructed as follows. For every $i \notin \{j, k, l\}$, set $P''_i = \hat{P}_i$. Let $P''_k = \bar{h}h_j h_k \dots$, and let P''_l be any element of $\mathbb{P}_l$. Again, by individual rationality, $\phi_i(P''_N) = h_i$ for all $i \notin \{j, k, l\}$. Since $P''_j = \bar{h}h_k \dots$ and $P''_k = \bar{h}h_j h_k \dots$, Pareto optimality implies that $\phi_j(P''_N) \neq h_j$. Indeed, if $\phi_j(P''_N) = h_j$, then necessarily $\phi_k(P''_N) = h_k$, and agents j and k could exchange their assignments and both become strictly better off, which contradicts Pareto optimality. Thus $\phi_j(P''_N) \in \{h_k, h_l\}$.

It follows that $\phi_j(P''_N)$ is strictly preferred to $\phi_j(P'_N) = h_j$ according to P'_j. Therefore the edge a prescribed by the truthful strategy at preference P'_j cannot be obviously dominant. Since P''_j was chosen arbitrarily in $\mathbb{P}_j$, the set $\mathbb{P}_j$ cannot be separated across distinct outgoing edges at node v. Hence there exists an edge $a \in O(v)$ such that $\mathbb{P}_j \subseteq \mathcal{R}_j(v, a)$.

Finally, for every $i \in N \setminus \{j\}$, we have $\mathcal{R}_i(v, a) = \mathcal{R}_i(v)$. Therefore $\mathbb{P}_N \subseteq \mathcal{R}_N(v, a)$, which completes the proof. ■

By definition, we have $\mathbb{P}_N \subseteq \mathcal{R}_N(v_0)$. By Claim .2.1, there exists an edge $a \in O(v_0)$ such that $\mathbb{P}_N \subseteq \mathcal{R}_N(v_0, a)$. Proceeding inductively in this manner along a path of the mechanism, we obtain $\mathbb{P}_N \subseteq \mathcal{R}_N(z)$ for some terminal node $z \in Z$.

Recall that there exist two distinct preference profiles in $\mathbb{P}_N$, namely P'_N and P''_N, such that $\phi(P'_N) \neq \phi(P''_N)$. Since both P'_N and P''_N belong to $\mathcal{R}_N(z)$ for the same terminal node $z \in Z$, it follows that the gradual revelation mechanism Γ assigns the same outcome at z to both profiles. Therefore, Γ cannot implement ϕ, which yields a contradiction. This completes the proof of the theorem.

.3 Proof of Theorem 6.4.5

We structure the proof into three parts. In Part I, we construct an extensive game form for the implementation of the TN algorithm. In Part II, we define a strategy profile in the game form that implements the TN algorithm. Finally, in Part III, we show that the strategy of each player defined in Part II is obviously dominant.

Part I of the proof : We define an extensive game form $\mathbb{E}$ through a recursive construction of a directed tree, denoted by $\mathcal{T}$, whose nodes are indexed by the pairs $(s_1, \ldots, s_k)$ and $(\underset{\sim}{i}^1, \ldots, \underset{\sim}{i}^k)$, where $k \in \mathbb{N}$, $s_l \in \{1, \ldots, n\}$ for each $l = 1, \ldots, k$, and each $\underset{\sim}{i}^l$ is a finite sequence consisting of the numbers 1 and 2. Each node is generically denoted by $v_{(s_1,\ldots,s_k)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}$. To align the structure with the TN algorithm, we refer to k as the Stage and s as the Step of the node.

Each node $v_{(s_1,\ldots,s_k)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}$ is associated with a label

$$\left(N_{(s_1,\ldots,s_k)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}, H_{(s_1,\ldots,s_k)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}, e_{(s_1,\ldots,s_k)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}, \Pi_{(s_1,\ldots,s_k)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}, \gamma_{(s_1,\ldots,s_k)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}\right),$$

where $N_{(s_1,\ldots,s_k)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}$ and $H_{(s_1,\ldots,s_k)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}$ are subsets of agents and houses, respectively, having the same cardinality; $e_{(s_1,\ldots,s_k)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}$ is a matching from the set $N_{(s_1,\ldots,s_k)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}$ to $H_{(s_1,\ldots,s_k)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}$; $\Pi_{(s_1,\ldots,s_k)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}$ is a subset of $H_{(s_1,\ldots,s_k)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}$; and $\gamma_{(s_1,\ldots,s_k)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}$ is an agent in $N_{(s_1,\ldots,s_k)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}$. We denote by $o_{(s_1,\ldots,s_k)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}$ the inverse of the function $e_{(s_1,\ldots,s_k)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}$. That is, $o_{(s_1,\ldots,s_k)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}$ is the ownership function with respect to the endowment function $e_{(s_1,\ldots,s_k)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}$.

For notational convenience, we sometimes denote a generic node by v instead of $v_{(s_1,\ldots,s_k)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}$ and denote the sets

$$N_{(s_1,\ldots,s_k)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}, \quad H_{(s_1,\ldots,s_k)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}, \quad e_{(s_1,\ldots,s_k)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}, \quad \Pi_{(s_1,\ldots,s_k)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}, \quad \gamma_{(s_1,\ldots,s_k)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}$$

by $N(v)$, $H(v)$, $e(v)$, $\Pi(v)$, and $\gamma(v)$, respectively. Throughout the proof, the root node $v_{(1)}^{(\underset{\sim}{i}^1)}$ is alternatively denoted by v_0. Each edge (v, v') is labeled with a house, which we denote by $L\big((v, v')\big)$.

We define the labels $L\big((v, v')\big)$ of all outgoing edges (v, v') from a node v recursively and simultaneously. Each such label is a house in $\mathrm{Nbr}\big(\Pi(v), H(v)\big)$. The notion of side consistency is used to determine these labels.

Given the labels of every edge on the path from the root v_0 to a node v, we say that a house $h \in \mathrm{Nbr}\big(\Pi(v), H(v)\big)$ is **side consistent** at v with player assignment $\gamma(v) = i$, for some $i \in N(v)$, if the following conditions hold:

- If $h \prec e(v)(i)$, then for every node v_u on the path $[v_1, \ldots, v_t]$ from the root $v_1 = v_0$ to $v_t = v$ such that $\gamma(v_u) = i$, it holds that

$$L\big((v_u, v_{u+1})\big) \prec e(v_u)(i).$$

- If $e(v)(i) \prec h$, then for every such node v_u, it holds that

$$e(v_u)(i) \prec L\big((v_u, v_{u+1})\big).$$

Observe that if there is no node on the path $[v_1, \ldots, v_t]$ with the same player assignment as node v, then every house at node v is side consistent. We denote the set of side consistent houses at node v by $\widehat{\mathrm{Nbr}}\big(\Pi(v), H(v)\big)$.

The tree $\mathcal{T}$ has as many outgoing edges from v as the number of elements in $\widehat{\mathrm{Nbr}}\big(\Pi(v), H(v)\big)$. Each outgoing edge (v, v') is labeled with a distinct, arbitrarily chosen house from $\widehat{\mathrm{Nbr}}\big(\Pi(v), H(v)\big)$.

We now introduce additional terminology before defining the tree $\mathcal{T}$. For any set of houses $\bar{H} \subsetneq \widehat{H} \subseteq H$, we define the set of border houses of $\bar{H}$ in $\widehat{H}$ as

$$B(\bar{H}, \widehat{H}) = \mathrm{Nbr}(\bar{H}, \widehat{H}) \setminus \bar{H}.$$

We place a hat over the notation of a set of houses to denote its subset consisting of side consistent houses. In particular, we use the notations $\widehat{\mathrm{Nbr}}\big(\Pi(v), H(v)\big)$, $\widehat{\Pi}(v)$, and $\widehat{B}\big(\Pi(v), H(v)\big)$ to denote the side consistent subsets of $\mathrm{Nbr}\big(\Pi(v), H(v)\big)$, $\Pi(v)$, and $B\big(\Pi(v), H(v)\big)$, respectively.

We now proceed to define the tree $\mathcal{T}$.

Step 1 of Stage 1. The root node, referred to as Step 1 of Stage 1, is denoted by $v_{(1)}^{(i_1^1)}$. Its associated labels are defined as

$$N_{(1)}^{(i_1^1)} = N, \qquad H_{(1)}^{(i_1^1)} = H, \qquad e_{(1)}^{(i_1^1)} = e,$$

$$\Pi_{(1)}^{(i_1^1)} = \left\{\tau\big(\prec, H_{(1)}^{(i_1^1)}\big)\right\}, \qquad \gamma_{(1)}^{(i_1^1)} = o_{(1)}^{(i_1^1)}\Big(\tau\big(\prec, H_{(1)}^{(i_1^1)}\big)\Big).$$

In view of the definition of side consistency, $\widehat{\Pi}_{(1)}^{(i_1^1)} = \Pi_{(1)}^{(i_1^1)}$. Let the unique house in $\widehat{\Pi}_{(1)}^{(i_1^1)}$ be denoted by $h_{(1)}^{(i_1^1)}$. The houses in $\widehat{B}\big(\Pi_{(1)}^{(i_1^1)}, H_{(1)}^{(i_1^1)}\big)$ are indexed arbitrarily as $h_{(2)}^{(i_1^1,1)}$ and $h_{(2)}^{(i_1^1,2)}$.

Depending on the values of $\Pi_{(1)}^{(i_1^1)}$ and $H_{(1)}^{(i_1^1)}$, there are at most two houses, and possibly none, in this set.

Step 2 of Stage 1. For each house $h_{(2)}^{(i_1^1,i_2^1)}$, where $i_2^1 \in \{1,2\}$, an edge is drawn from $v_{(1)}^{(i_1^1)}$ to a node at Step 2 of Stage 1, denoted by $v_{(2)}^{(i_1^1,i_2^1)}$. The labels of this node are defined as

$$\begin{aligned}
N_{(2)}^{(i_1^1,i_2^1)} &= N_{(1)}^{(i_1^1)},\\
H_{(2)}^{(i_1^1,i_2^1)} &= H_{(1)}^{(i_1^1)},\\
e_{(2)}^{(i_1^1,i_2^1)} &= e_{(1)}^{(i_1^1)},\\
\Pi_{(2)}^{(i_1^1,i_2^1)} &= \Pi_{(1)}^{(i_1^1)} \cup \left\{h_{(2)}^{(i_1^1,i_2^1)}\right\},\\
\gamma_{(2)}^{(i_1^1,i_2^1)} &= o_{(2)}^{(i_1^1,i_2^1)}\left(h_{(2)}^{(i_1^1,i_2^1)}\right).
\end{aligned}$$

For a node $v_{(2)}^{(i_1^1,i_2^1)}$, the houses in $\widehat{B}\big(\Pi_{(2)}^{(i_1^1,i_2^1)}, H_{(2)}^{(i_1^1,i_2^1)}\big)$ are indexed as $h_{(3)}^{(i_1^1,i_2^1,1)}$ and $h_{(3)}^{(i_1^1,i_2^1,2)}$. Again, there are at most two houses, and possibly none, in this set.

We now describe a generic step in Stage 1.

Step s_1 of Stage 1. Consider a node $v_{(s_1-1)}^{(i_1^1,\ldots,i_{s_1-1}^1)}$ at Step $s_1 - 1$ of Stage 1. For each house $h_{(s_1)}^{(i_1^1,\ldots,i_{s_1}^1)}$ in the set $\widehat{B}\big(\Pi_{(s_1-1)}^{(i_1^1,\ldots,i_{s_1-1}^1)}, H_{(s_1-1)}^{(i_1^1,\ldots,i_{s_1-1}^1)}\big)$, an edge is drawn to a node at Step s_1 of Stage 1, denoted by $v_{(s_1)}^{(i_1^1,\ldots,i_{s_1}^1)}$.

The labels of this node are defined as

$$\begin{aligned}
N_{(s_1)}^{(i_1^1,\ldots,i_{s_1}^1)} &= N_{(s_1-1)}^{(i_1^1,\ldots,i_{s_1-1}^1)},\\
H_{(s_1)}^{(i_1^1,\ldots,i_{s_1}^1)} &= H_{(s_1-1)}^{(i_1^1,\ldots,i_{s_1-1}^1)},\\
e_{(s_1)}^{(i_1^1,\ldots,i_{s_1}^1)} &= e_{(s_1-1)}^{(i_1^1,\ldots,i_{s_1-1}^1)},\\
\Pi_{(s_1)}^{(i_1^1,\ldots,i_{s_1}^1)} &= \Pi_{(s_1-1)}^{(i_1^1,\ldots,i_{s_1-1}^1)} \cup \left\{h_{(s_1)}^{(i_1^1,\ldots,i_{s_1}^1)}\right\},\\
\gamma_{(s_1)}^{(i_1^1,\ldots,i_{s_1}^1)} &= o_{(s_1)}^{(i_1^1,\ldots,i_{s_1}^1)}\left(h_{(s_1)}^{(i_1^1,\ldots,i_{s_1}^1)}\right).
\end{aligned}$$

Since the set $\Pi_{(s_1)}^{(i_1^1,\ldots,i_{s_1}^1)}$ strictly increases with the length of the sequence $(i_1^1, \ldots, i_{s_1}^1)$, the set $\widehat{B}\big(\Pi_{(s_1)}^{(i_1^1,\ldots,i_{s_1}^1)}, H_{(s_1)}^{(i_1^1,\ldots,i_{s_1}^1)}\big)$ becomes empty after finitely many steps. Consequently, the construction of Stage 1 terminates after a finite number of steps.

Next, we define Stage 2. We first present Step 1 of this stage and then describe a generic step.

Step 1 of Stage 2: We first construct the nodes of the tree corresponding to Step 1 of Stage 2. Consider an arbitrary node $v_{(s_1)}^{(i_1^1,\ldots,i_{s_1}^1)}$ in Step s_1 of Stage 1. From this node, we draw a number of edges to nodes in Step 1 of Stage 2. From the structure of the labels defined in Stage 1, it follows that the houses in $\Pi_{(s_1)}^{(i_1^1,\ldots,i_{s_1}^1)}$ corresponding to the node $v_{(s_1)}^{(i_1^1,\ldots,i_{s_1}^1)}$ are precisely

$$\left\{h_{(1)}^{(i_1^1)},\ h_{(2)}^{(i_1^1,i_2^1)},\ \ldots,\ h_{(s_1)}^{(i_1^1,\ldots,i_{s_1}^1)}\right\}.$$

For each house $h_{(t)}^{(i_1^1,\ldots,i_t^1)} \in \widehat{\Pi}_{(s_1)}^{(i_1^1,\ldots,i_{s_1}^1)} \subseteq \left\{h_{(1)}^{(i_1^1)},\ h_{(2)}^{(i_1^1,i_2^1)},\ \ldots,\ h_{(s_1)}^{(i_1^1,\ldots,i_{s_1}^1)}\right\}$, we draw one edge to a node in Stage 2, denoted by $v_{(s_1,1)}^{(i_1^1,\ldots,i_t^1),(i_1^2)}$, with label

$$\left(N_{(s_1,1)}^{(i_1^1,\ldots,i_t^1),(i_1^2)},\ H_{(s_1,1)}^{(i_1^1,\ldots,i_t^1),(i_1^2)},\ e_{(s_1,1)}^{(i_1^1,\ldots,i_t^1),(i_1^2)},\ \Pi_{(s_1,1)}^{(i_1^1,\ldots,i_t^1),(i_1^2)},\ \gamma_{(s_1,1)}^{(i_1^1,\ldots,i_t^1),(i_1^2)}\right).$$

These components are defined as follows:

$$\begin{aligned}
N_{(s_1,1)}^{(i_1^1,\ldots,i_t^1),(i_1^2)} &= N_{(s_1)}^{(i_1^1,\ldots,i_{s_1}^1)} \setminus \gamma_{(s_1)}^{(i_1^1,\ldots,i_{s_1}^1)}, \\
H_{(s_1,1)}^{(i_1^1,\ldots,i_t^1),(i_1^2)} &= H_{(s_1)}^{(i_1^1,\ldots,i_{s_1}^1)} \setminus \{h_{(t)}^{(i_1^1,\ldots,i_t^1)}\}, \\
e_{(s_1,1)}^{(i_1^1,\ldots,i_t^1),(i_1^2)}(i) &= \begin{cases} h_{(u+1)}^{(i_1^1,\ldots,i_{u-1}^1,i_u^1)} & \text{if } i = \gamma_{(u)}^{(i_1^1,\ldots,i_{u-1}^1)},\ t \le u < s_1, \\ e_{(s_1)}^{(i_1^1,\ldots,i_{t-1}^1)}(i) & \text{otherwise}, \end{cases} \\
\Pi_{(s_1,1)}^{(i_1^1,\ldots,i_t^1),(i_1^2)} &= \left\{\tau\big(\prec, H_{(s_1,1)}^{(i_1^1,\ldots,i_t^1),(i_1^2)}\big)\right\}, \\
\gamma_{(s_1,1)}^{(i_1^1,\ldots,i_t^1),(i_1^2)} &= o_{(s_1,1)}^{(i_1^1,\ldots,i_t^1),(i_1^2)}\left(\tau\big(\prec, H_{(s_1,1)}^{(i_1^1,\ldots,i_t^1),(i_1^2)}\big)\right).
\end{aligned}$$

Observe that the agent $\gamma_{(s_1)}^{(i_1^1,\ldots,i_{s_1}^1)}$ and the house $h_{(t)}^{(i_1^1,\ldots,i_t^1)}$ are removed during the transition from Step s_1 of Stage 1 to Step 1 of Stage 2. For clarity, we state that the pair

$$\left(\gamma_{(s_1)}^{(i_1^1,\ldots,i_{s_1}^1)},\ h_{(t)}^{(i_1^1,\ldots,i_t^1)}\right)$$

is matched and removed at the end of Stage 1.

$$\vdots$$

Step s_2 of Stage 2: Consider an arbitrary node $v_{(s_1,s_2-1)}^{(i_1^1,\ldots,i_{s_1}^1),(i_1^2,\ldots,i_{s_2-1}^2)}$ in Step s_2-1 of Stage 2 of the tree $\mathcal{T}$. Let the houses in the set $\widehat{B}\left(\Pi_{(s_1,s_2-1)}^{(i_1^1,\dots,i_{s_1}^1),(i_1^2,\dots,i_{s_2-1}^2)}, H_{(s_1,s_2-1)}^{(i_1^1,\dots,i_{s_1}^1),(i_1^2,\dots,i_{s_2-1}^2)}\right)$ be indexed as

$$h_{(s_1,s_2)}^{(i_1^1,\dots,i_{s_1}^1),(i_1^2,\dots,i_{s_2-1}^2,1)} \quad \text{and} \quad h_{(s_1,s_2)}^{(i_1^1,\dots,i_{s_1}^1),(i_1^2,\dots,i_{s_2-1}^2,2)}.$$

For each such house $h_{(s_1,s_2)}^{(i_1^1,\dots,i_{s_1}^1),(i_1^2,\dots,i_{s_2}^2)}$, an edge is drawn to a node at Step s_2 of Stage 2, denoted by $v_{(s_1,s_2)}^{(i_1^1,\dots,i_{s_1}^1),(i_1^2,\dots,i_{s_2}^2)}$. The label of this node is defined as follows:

$$\begin{aligned}
N_{(s_1,s_2)}^{(i_1^1,\dots,i_{s_1}^1),(i_1^2,\dots,i_{s_2}^2)} &= N_{(s_1,s_2-1)}^{(i_1^1,\dots,i_{s_1}^1),(i_1^2,\dots,i_{s_2-1}^2)}, \\
H_{(s_1,s_2)}^{(i_1^1,\dots,i_{s_1}^1),(i_1^2,\dots,i_{s_2}^2)} &= H_{(s_1,s_2-1)}^{(i_1^1,\dots,i_{s_1}^1),(i_1^2,\dots,i_{s_2-1}^2)}, \\
e_{(s_1,s_2)}^{(i_1^1,\dots,i_{s_1}^1),(i_1^2,\dots,i_{s_2}^2)} &= e_{(s_1,s_2-1)}^{(i_1^1,\dots,i_{s_1}^1),(i_1^2,\dots,i_{s_2-1}^2)}, \\
\Pi_{(s_1,s_2)}^{(i_1^1,\dots,i_{s_1}^1),(i_1^2,\dots,i_{s_2}^2)} &= \Pi_{(s_1,s_2-1)}^{(i_1^1,\dots,i_{s_1}^1),(i_1^2,\dots,i_{s_2-1}^2)} \cup \left\{h_{(s_1,s_2)}^{(i_1^1,\dots,i_{s_1}^1),(i_1^2,\dots,i_{s_2}^2)}\right\}, \\
\gamma_{(s_1,s_2)}^{(i_1^1,\dots,i_{s_1}^1),(i_1^2,\dots,i_{s_2}^2)} &= o_{(s_1,s_2)}^{(i_1^1,\dots,i_{s_1}^1),(i_1^2,\dots,i_{s_2}^2)}\left(h_{(s_1,s_2)}^{(i_1^1,\dots,i_{s_1}^1),(i_1^2,\dots,i_{s_2}^2)}\right).
\end{aligned}$$

$$\vdots$$

Step s_k of Stage k: Now consider an arbitrary node $v_{(s_1,\dots,s_k-1)}^{(\underset{\sim}{i}^1,\dots,\dots,\underset{\sim}{i}^k)}$ in Step s_k-1 of Stage k of the tree $\mathcal{T}$. Let the houses in the set $\widehat{B}\big(\Pi_{(s_1,\dots,s_k-1)}^{(\underset{\sim}{i}^1,\dots,\dots,\underset{\sim}{i}^k)}, H_{(s_1,\dots,s_k-1)}^{(\underset{\sim}{i}^1,\dots,\dots,\underset{\sim}{i}^k)}\big)$ be indexed as

$$h_{(s_1,\dots,s_k)}^{(\underset{\sim}{i}^1,\dots,\dots,\underset{\sim}{i}^{k-1},(i_1^k,\dots,i_{s_k-1}^k,1))} \text{ and } h_{(s_1,\dots,s_k)}^{(\underset{\sim}{i}^1,\dots,\dots,\underset{\sim}{i}^{k-1},(i_1^k,\dots,i_{s_k-1}^k,2))}.$$

For each such house $h_{(s_1,\dots,s_k)}^{(\underset{\sim}{i}^1,\dots,\dots,\underset{\sim}{i}^{k-1},(i_1^k,\dots,i_{s_k-1}^k,i_{s_k}^k))}$, we draw an edge to a node at the level Step s_k of Stage k, denoted by $v_{(s_1,\dots,s_k)}^{(\underset{\sim}{i}^1,\dots,\dots,\underset{\sim}{i}^{k-1},(i_1^k,\dots,i_{s_k-1}^k,i_{s_k}^k))}$.

The label of this node is defined as follows:

$$
\begin{aligned}
N_{(s_1,\ldots,s_k)}^{(\utilde{i}^1,\ldots,\ldots,\utilde{i}^{k-1},(i_1^k,\ldots,i_{s_k-1}^k,i_{s_k}^k))} &= N_{(s_1,\ldots,s_k-1)}^{(\utilde{i}^1,\ldots,\ldots,\utilde{i}^{k-1},(i_1^k,\ldots,i_{s_k-1}^k))}, \\
H_{(s_1,\ldots,s_k)}^{(\utilde{i}^1,\ldots,\ldots,\utilde{i}^{k-1},(i_1^k,\ldots,i_{s_k-1}^k,i_{s_k}^k))} &= H_{(s_1,\ldots,s_k-1)}^{(\utilde{i}^1,\ldots,\ldots,\utilde{i}^{k-1},(i_1^k,\ldots,i_{s_k-1}^k))}, \\
e_{(s_1,\ldots,s_k)}^{(\utilde{i}^1,\ldots,\ldots,\utilde{i}^{k-1},(i_1^k,\ldots,i_{s_k-1}^k,i_{s_k}^k))} &= e_{(s_1,\ldots,s_k-1)}^{(\utilde{i}^1,\ldots,\ldots,\utilde{i}^{k-1},(i_1^k,\ldots,i_{s_k-1}^k))}, \\
\Pi_{(s_1,\ldots,s_k)}^{(\utilde{i}^1,\ldots,\ldots,\utilde{i}^{k-1},(i_1^k,\ldots,i_{s_k-1}^k,i_{s_k}^k))} &= \Pi_{(s_1,\ldots,s_k-1)}^{(\utilde{i}^1,\ldots,\ldots,\utilde{i}^{k-1},(i_1^k,\ldots,i_{s_k-1}^k))} \cup \left\{ h_{(s_1,\ldots,s_k)}^{(\utilde{i}^1,\ldots,\ldots,\utilde{i}^{k-1},(i_1^k,\ldots,i_{s_k-1}^k,i_{s_k}^k))} \right\}, \\
\gamma_{(s_1,\ldots,s_k)}^{(\utilde{i}^1,\ldots,\ldots,\utilde{i}^{k-1},(i_1^k,\ldots,i_{s_k-1}^k,i_{s_k}^k))} &= o_{(s_1,\ldots,s_k)}^{(\utilde{i}^1,\ldots,\ldots,\utilde{i}^{k-1},(i_1^k,\ldots,i_{s_k-1}^k,i_{s_k}^k))} \left(h_{(s_1,\ldots,s_k)}^{(\utilde{i}^1,\ldots,\ldots,\utilde{i}^{k-1},(i_1^k,\ldots,i_{s_k-1}^k,i_{s_k}^k))} \right).
\end{aligned}
$$

The algorithm terminates in Stage $n+1$. At the end of each stage, exactly one agent exits the algorithm with a house. Consequently, by Stage $n+1$ only a single agent remains. Hence, the algorithm terminates at the first step of Stage $n+1$.

This completes the description of the extensive game form $\mathbb{E}$. Before proceeding to the subsequent parts of the proof, we state and establish certain properties of $\mathbb{E}$ that will be essential for the remainder of the argument.

Claim .3.1 *The set $\widehat{\Pi}(v)$ at any node v forms an interval in $H(v)$. More precisely, for $h, h', h'' \in H(v)$ with $h \prec h' \prec h''$, we have $h, h'' \in \widehat{\Pi}(v)$ implies $h' \in \widehat{\Pi}(v)$.*

Proof: We prove the claim by induction on the nodes of the tree $\mathcal{T}$. For the root node v_0, the set $\widehat{\Pi}(v_0)$ is a singleton consisting of the house owned by the agent $\gamma(v_0)$. Hence, it is trivially an interval in $H(v_0)$.

Assume that $\widehat{\Pi}(v)$ is an interval in $H(v)$ for some node v. Since the underlying graph is a tree, it suffices to verify that the property holds for each child of v.[8]

Let v' be a child of v, and let the label of the edge (v, v'), denoted by $L((v, v'))$, be h, where $h \in \widehat{\mathrm{Nbr}}(\Pi(v), H(v))$. Recall that $\widehat{\mathrm{Nbr}}(\Pi(v), H(v)) = \widehat{\Pi}(v) \cup \widehat{B}(\Pi(v), H(v))$. We distinguish two cases depending on whether $h \in \widehat{\Pi}(v)$ or $h \in \widehat{B}(\Pi(v), H(v))$.

Case 1: $h \in \widehat{\Pi}(v)$.

By the definition of the extensive game form $\mathbb{E}$, in this case $\widehat{\Pi}(v')$ is a singleton consisting of the house $\tau(\prec, H(v'))$. Hence, $\widehat{\Pi}(v')$ is trivially an interval in $H(v')$.

[8]A node v' is said to be a child of a node v in the tree $\mathcal{T}$ if there exists a directed edge (v, v') in $\mathcal{T}$.

Case 2: $h \in \widehat{B}(\Pi(v), H(v))$.

Then $\widehat{\Pi}(v') = \widehat{\Pi}(v) \cup \{h\}$. By the definition of $\widehat{B}(\Pi(v), H(v))$, the house h is a neighbor of the interval $\widehat{\Pi}(v)$ in $H(v)$. Therefore, the set $\widehat{\Pi}(v) \cup \{h\}$ forms an interval in $H(v)$. Moreover, when $h \in \widehat{B}(\Pi(v), H(v))$, we have $H(v') = H(v)$ by definition. Hence, $\widehat{\Pi}(v')$ is an interval in $H(v')$.

In both cases, $\widehat{\Pi}(v')$ is an interval in $H(v')$. This completes the induction and establishes the claim. $\blacksquare$

Part II of the proof : In this part of the proof, we construct a strategy profile that implements the TN algorithm.

For every agent $i \in N$, we define the strategy σ_i such that for every $P_i \in \mathcal{S}(T)$ and at every node v with $\gamma(v) = i$, $\sigma_i(P_i)(v) = \tau\big(P_i, O(v)\big)$ where $O(v)$ is the set of outgoing edges from the node v. Since at every node, the assigned player chooses a unique edge from its outgoing edges, the strategy-profiles $\sigma_N = \prod_{i \in N} \sigma_i$ will lead to a terminal node of the tree $\mathcal{T}$. Let $[v_{(1)}^{(\underset{\sim}{i}^1)}, \ldots, v_{(s_1,\ldots,s_n,1)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^{n+1})}]$ be the path induced by the strategy-profile s_N from the root $v_{(1)}^{(\underset{\sim}{i}^1)}$ to the terminal node $v_{(s_1,\ldots,s_n,1)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^{n+1})}$ in Step 1 of Stage $n+1$. Let $v_{(1)}^{(\underset{\sim}{i}^1)}, v_{(s_1,1)}^{(\underset{\sim}{i}^1,\underset{\sim}{i}^2)}, \ldots, v_{(s_1,\ldots,s_n,1)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^{n+1})}$ be all the nodes in the path that belong to the first step in some stage. We match the house $H_{(s_1,\ldots,s_{l-1},1)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^l)} \setminus H_{(s_1,\ldots,s_l,1)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^{l+1})}$ with the agent $N_{(s_1,\ldots,s_{l-1},1)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^l)} \setminus N_{(s_1,\ldots,s_l,1)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^{l+1})}$ for each $l \in \{1, \ldots, n\}$. We denote this matching by $\mu(s_N(P_N))$. Note that by the definition of the extensive game form $\mathbb{E}$, $H_{(s_1,\ldots,s_n,1)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^{n+1})}$ and $N_{(s_1,\ldots,s_l,1)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^{l+1})}$ are empty. This implies that $\mu(s_N(P_N))$ is indeed a matching.

Now, we show that the strategy defined above implements the desired outcome. Consider an arbitrary preference profile P_N. We demonstrate that the flow of the game in the tree under the given strategy profile coincides exactly with the execution of the TN algorithm. In particular, we establish that at the end of each stage, the same pair consisting of an agent and a house is matched in both the game form and the algorithm. This is shown by induction on the stages of the process.

Induction Hypothesis: Suppose (i) the extensive game form starts at the node $v_{(s_1,\ldots,s_{k-1},1)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}$ at Stage k with labels

$$\big(N_{(s_1,\ldots,s_{k-1},1)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}, H_{(s_1,\ldots,s_{k-1},1)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}, e_{(s_1,\ldots,s_{k-1},1)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}, \Pi_{(s_1,\ldots,s_{k-1},1)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}, \gamma_{(s_1,\ldots,s_{k-1},1)}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}\big)$$

and the TN algorithm starts at Stage k with the parameters

$$N_k, H_k, e_k, \Pi(k,1), ando_k(\pi(k,1)).$$

Assume that $N_k = N^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},1)}$, $H_k = H^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},1)}$, $e_k = e^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},1)}$, $\Pi(k,1) = \Pi^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},1)}$, and $o_k(\pi(k,1)) = \gamma^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},1)}$.

Induction Step. We aim to establish the following two claims:

1. The same pair of agent and house will be matched in both the extensive game form and the TN algorithm.

2. The parameters of the node reached at the beginning of Stage $k+1$ in the extensive game form will be equivalent to those in the corresponding state at the beginning of Stage $k+1$ in the TN algorithm. That is, if the extensive game form reaches the node $v^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^{k+1})}_{(s_1,\ldots,s_k,1)}$ with label

$$\left(N^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^{k+1})}_{(s_1,\ldots,s_k,1)},\ H^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^{k+1})}_{(s_1,\ldots,s_k,1)},\ e^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^{k+1})}_{(s_1,\ldots,s_k,1)},\ \Pi^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^{k+1})}_{(s_1,\ldots,s_k,1)},\ \gamma^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^{k+1})}_{(s_1,\ldots,s_k,1)}\right),$$

and the TN algorithm reaches a state with parameters

$$N_{k+1}, H_{k+1}, e_{k+1}, \Pi(k+1,1), \text{ and } o_{k+1}(\pi(k+1,1))$$

at the beginning of Stage $k+1$, then the following equalities hold:

$$N_{k+1} = N^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^{k+1})}_{(s_1,\ldots,s_{k+1})}, \quad H_{k+1} = H^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^{k+1})}_{(s_1,\ldots,s_{k+1})}, \quad e_{k+1} = e^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^{k+1})}_{(s_1,\ldots,s_{k+1})},$$

$$\Pi(k+1,1) = \Pi^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^{k+1})}_{(s_1,\ldots,s_{k+1})}, \quad o_{k+1}(\pi(k+1,1)) = \gamma^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^{k+1})}_{(s_1,\ldots,s_{k+1})}.$$

We begin by establishing the base case.

Base Case (Stage 1): The game starts at the root node $v^{i^1_1}_{(1)}$ at the beginning of Stage 1. By definition, we have $N^{i^1_1}_{(1)} = N_1 (= N)$, $H^{i^1_1}_{(1)} = H_1 (= H)$, $e^{i^1_1}_{(1)} = e_1 (= e)$, $\Pi^{(i^1_1)}_{(1)} = \Pi(1,1) (= \{\tau(\prec, H\})$, and $\gamma^{i^1_1}_{(1)} = o_1(\pi(1,1)) \big(= \tau(\prec, H)\big)$. This shows that the induction hypothesis holds for the base case. We now proceed to show a general induction step.

Consider an arbitrary Stage k. Suppose that the induction hypothesis holds, that is, $N_k = N^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},1)}$, $H_k = H^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},1)}$, $e_k = e^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},1)}$, $\Pi(k,1) = \Pi^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},1)}$, and $o_k(\pi(k,1)) = \gamma^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},1)}$. We prove the induction step using another layer of induction over the steps in Stage k. To ease the presentation of the proof, for the TN algorithm, let us define $N_{k,t} = N_k$, $H_{k,t} = H_k$, and $e_{k,t} = e_k$ for all Steps t in Stage k. We further drop the superscript and all but the last element in the subscript in the notations related to the extensive game form. For instance, we denote $N^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},1)}$, $H^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},1)}$, and $e^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},1)}$ by N_1, H_1, and e_1 respectively.

Consider an arbitrary Step s in Stage k. Suppose the parameters of TN algorithm and the corresponding node in the extensive game tree $\mathcal{T}$ match at the beginning of Step s, that is,

$$\begin{aligned} N_{k,s} &= N_s \\ H_{k,s} &= H_s \\ e_{k,s} &= e_s \\ \Pi(k,s) &= \Pi_s \\ o_k(\pi(k,s)) &= \gamma_s \end{aligned} \tag{2}$$

We show that they will match at the beginning of Step $s+1$.

Note that from the definition of TN algorithm and extensive game form $\mathbb{E}$, the set of remaining agents, remaining houses, and current endowments do not change over the steps for both of them. Since, at the beginning of Step s, each of them were equal across the TN algorithm and the single-peaked game form, we have $N_{k,s+1} = N_{s+1}$, $H_{k,s+1} = H_{s+1}$, $e_{k,s+1} = e_{s+1}$.

We next show that the owner $o_k\big(\tau(P_{o_k(\pi(k,s))},\ \mathrm{Nbr}\,(\Pi(k,s), H_k))\big)$ of the pointer $\tau(P_{o_k(\pi(k,s))},\ \mathrm{Nbr}\,(\Pi(k,s), H_k))$ at Step $s+1$ in TN algorithm is the same as the player assigned to the unique node, say v_{s+1}, where the game progresses according to the profile P_N from the current node v_s. By the definition of $\mathbb{E}$, the player assigned to the node v_{s+1} at the strategy-profile σ_N at the preference profile P_N is $o_s\big(\tau(P_{\gamma_s},\ \widehat{\mathrm{Nbr}}\,(\Pi_s, H_s)\big)$. Using the equalities from Equation 2, to show

$$o_k\big(\tau(P_{o_k(\pi(k,s))},\ \mathrm{Nbr}\,(\Pi(k,s), H_k))\big) = o_s\big(\tau(P_{\gamma_s},\ \widehat{\mathrm{Nbr}}\,(\Pi_s, H_s)\big),$$

it suffices to show that

$$\tau(P_{\gamma_s},\ \mathrm{Nbr}\,(\Pi_s, H_s)) = \tau(P_{\gamma_s},\ \widehat{\mathrm{Nbr}}\,(\Pi_s, H_s)). \tag{3}$$

Assume wlog that $\tau(P_{\gamma_s},\ \mathrm{Nbr}\,(\Pi_s, H_s)) \preceq e_s(\gamma_s)$. It is sufficient to show that there does not exist any

$h \in \mathrm{Nbr}\,(\Pi_s, H_s) \setminus \widehat{\mathrm{Nbr}}\,(\Pi_s, H_s)$ such that $h \preceq e_s(\gamma_s)$, since this, together with our assumption $\tau(P_{\gamma_s},\ \mathrm{Nbr}\,(\Pi_s, H_s)) \preceq e_s(\gamma_s)$, implies $\tau(P_{\gamma_s},\ \mathrm{Nbr}\,(\Pi_s, H_s)) = \tau(P_{\gamma_s},\ \widehat{\mathrm{Nbr}}\,(\Pi_s, H_s))$. Let $v_{j_1}, \ldots, v_{j_p}$ be the sequence of nodes along the path from the root node v_0 to the node v_s such that $\gamma(v_{j_r}) = \gamma_s$ for each $r \in \{1, \ldots, p\}$. Thus v_{j_1} is the first node along this path where agent γ_s takes an action for the first time, and $v_{j_p} = v_s$. It remains to show that $\sigma_{\gamma_s}(P_N)(v_{j_1}) \preceq e_{\gamma_s}(v_{j_1})$. This follows from the fact that, by the definition of the extensive game form $\mathbb{E}$ and Claim .3.1, $\tau(P_{\gamma_s},\ \widehat{\mathrm{Nbr}}\,(\Pi_{j_r}, H_{j_r})) \preceq e_{\gamma_s}(v_{j_r})$ for each $r \in \{1, \ldots, p\}$. Indeed, for the peak to cross from one side of the endowment to the other, there must exist a time at which the endowment itself coincides with the peak, in which case the agent would receive that house.

Comparing the two terms, we can see that $h = \pi(k, s+1)$. This implies that $o_k(\pi(k, s+1)) = \gamma^{(\underset{\sim}{i}^1, \ldots, \underset{\sim}{i}^k)}_{(s_1, \ldots, s_{k-1}, s+1)}$. Moreover, since $\Pi(k, s+1) = \Pi(k, s) \cup \pi(k, s+1)$ and $\Pi^{(\underset{\sim}{i}^1, \ldots, \underset{\sim}{i}^k)}_{(s_1, \ldots, s_{k-1}, s+1)} = \Pi^{(\underset{\sim}{i}^1, \ldots, \underset{\sim}{i}^k)}_{(s_1, \ldots, s_{k-1}, s)} \cup h$, we have $\Pi(k, s+1) = \Pi^{(\underset{\sim}{i}^1, \ldots, \underset{\sim}{i}^k)}_{(s_1, \ldots, s_{k-1}, s+1)}$. This concludes that the parameters related to Step $s+1$ in both TN algorithm and the extensive game form will be the same. Since at Step 1 of Stage k, the parameters of TN algorithm and single-peaked game form are the same, by induction we have the parameters will also be the same at the last step of Stage k, and hence in the beginning of Stage $k+1$.

Part III of the proof: In this part, we establish that the strategy profile σ_N is obviously strategy-proof (osp) in $\mathbb{E}$. In particular, consider any arbitrary preference profile (P_i, P_{-i}), and let $v^{(\underset{\sim}{i}^1, \ldots, \underset{\sim}{i}^k)}_{(s_1, \ldots, s_{k-1}, s)}$ be a node in Step s of Stage k that is reached when the strategy profile $(\sigma_i(P_i), \sigma_{-i}(P_{-i}))$ is followed. Then, we need to show that the action (i.e., edge) prescribed by the strategy of player $i = \gamma^{(\underset{\sim}{i}^1, \ldots, \underset{\sim}{i}^k)}_{(s_1, \ldots, s_{k-1}, s)}$ at this node is obviously dominant.

Let $H^{(\underset{\sim}{i}^1, \ldots, \underset{\sim}{i}^k)}_{(s_1, \ldots, s_{k-1}, s)} = \{h^1, \ldots, h^m\}$ with $h^1 \prec \cdots \prec h^m$. By Claim .3.1, $\widehat{\Pi}^{(\underset{\sim}{i}^1, \ldots, \underset{\sim}{i}^k)}_{(s_1, \ldots, s_{k-1}, s)}$ forms an interval in $H^{(\underset{\sim}{i}^1, \ldots, \underset{\sim}{i}^k)}_{(s_1, \ldots, s_{k-1}, s)}$. Suppose $\Pi^{(\underset{\sim}{i}^1, \ldots, \underset{\sim}{i}^k)}_{(s_1, \ldots, s_{k-1}, s)} = \{h^p, \ldots, h^q, \ldots, h^r\}$, where $1 \leq p < q < r \leq m$, and $e^{(\underset{\sim}{i}^1, \ldots, \underset{\sim}{i}^k)}_{(s_1, \ldots, s_{k-1}, s)}(i) = h^q$. By the definition of $\widehat{B}\big(\Pi^{(\underset{\sim}{i}^1, \ldots, \underset{\sim}{i}^k)}_{(s_1, \ldots, s_{k-1}, s)}, H^{(\underset{\sim}{i}^1, \ldots, \underset{\sim}{i}^k)}_{(s_1, \ldots, s_{k-1}, s)}\big)$, we have

$\widehat{B}\big(\Pi^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}, H^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}\big) = \{h^{p-1}, h^{r+1}\} \cap H^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}$. More precisely,

$$\widehat{B}\big(\Pi^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}, H^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}\big) = \begin{cases} \{h^{p-1},\ h^{r+1}\}, & \text{if } 1 < p < r < m, \\ \{h^{p-1}\}, & \text{if } p = 1 \text{ and } r < m, \\ \{h^{r+1}\}, & \text{if } p > 1 \text{ and } r = m. \end{cases} \tag{4}$$

We know that the set of edges outgoing from $v^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}$ is partitioned into two subsets:

$$\widehat{\Pi}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)} = \{h^p, \ldots, h^r\}$$

and

$$\widehat{B}\big(\Pi^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}, H^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}\big) = \{h^{p-1}, h^{r+1}\} \cap H^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}.$$

Let $\sigma_i(P_i)\big(v^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}\big) = h$. We distinguish two cases depending on whether h belongs to $\widehat{\Pi}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}$ or to $\widehat{B}\big(\Pi^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}, H^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}\big)$.

Case 1. Suppose $h \in \widehat{\Pi}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}$. By the definition of the extensive game form $\mathbb{E}$, agent i is assigned the house h. Moreover, again by the definition of $\mathbb{E}$, only houses in $H^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}$ can be allocated to agent i under any strategy.

We claim that h is the most preferred house of agent i within the set $H^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}$. Suppose, for the sake of contradiction, that there exists another house $h' \in H^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}$ such that $h' \succ_i h$. Without loss of generality, assume that $h^r \prec h'$.

By single-peakedness of preferences, it follows that the most preferred house of agent i in the set $\{h^{p-1}, \ldots, h^{r+1}\}$ must be h^{r+1}. This contradicts the fact that h is the most preferred element in $\widehat{\mathrm{Nbr}}\big(\Pi^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}, H^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}\big) = \{h^{p-1}, \ldots, h^{r+1}\}$. Hence, h is indeed the most preferred house for agent i in $H^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}$.

Case 2. Suppose $h \in \widehat{B}\big(\Pi^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}, H^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}\big)$. Without loss of generality, assume that $h = h^{r+1}$. Consider an alternative strategy σ'_i for agent i under which, at the node $v^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}$, the agent selects some house $h' \neq h$.

By the definition of the TN algorithm, if the node $v^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}$ is the first node (along the path from the root $v^{(i^1)}_{(1)}$) at which agent i takes an action, then $h' \in \widehat{\Pi}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)} \cup \{h^{p-1}\}$; otherwise, $h' \in \widehat{\Pi}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}$.

Since $\sigma_i(P_i)\big(v^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}\big) = h^{r+1}$, it follows from the definition of the extensive game form $\mathbb{E}$ that h^{r+1} is the most preferred house of agent i in the set $\{h^{p-1}, \ldots, h^{r+1}\}$.

We first establish the following claim.

Claim .3.2 $m(\sigma_i(P_i), \sigma_{-i}(P_{-i}))\, R_i\, h^r$.

Proof: We prove the claim using Lemma .1.3. For the convenience of the reader, we restate the lemma here. To avoid notational conflicts, we use slightly modified notation.

Statement of Lemma .1.3. Let $i' = o(\pi(k,s))$ for some Stage k and Step s in that stage. If $\tau(P_{i'}, H_k) \notin \Pi(k,s)$ and $x = \tau\big(P_{i'}, \mathrm{Nbr}(\Pi(k,s), H_k)\big)$, then

$$\bar{\phi}_{i'}(P_{i'}, P_{-i'})\, R_{i'}\, x',$$

where x' denotes the projection of x onto $\Pi(k,s)$.

By Part II of the proof of Theorem 6.4.5, we have

$$i' = i, \quad H_k = H^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}, \quad \text{and} \quad \Pi(k,s) = \Pi^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}.$$

We first show that $x = h^{r+1}$. By the definition of h^{r+1}, we have

$$h^{r+1} = \tau\Big(P_i, \widehat{\mathrm{Nbr}}\big(\Pi^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}, H^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}\big)\Big).$$

If $v^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}$ is the first node along the path from the root $v^{(i^1)}_{(1)}$ at which agent i takes an action, then

$$\widehat{\mathrm{Nbr}}\big(\Pi^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}, H^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}\big) = \mathrm{Nbr}\big(\Pi^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}, H^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}\big) = \{h^{p-1}, \ldots, h^{r+1}\},$$

which directly implies that $x = h^{r+1}$.

Otherwise, by side-consistency,

$$\widehat{\mathrm{Nbr}}\big(\Pi^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}, H^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}\big) = \{h^q, \ldots, h^{r+1}\}.$$

Since $\sigma_i(P_i)\big(v^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}\big) = h^{r+1}$, it follows that $h^{r+1} \preceq \tau\big(P_i, H^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}\big)$. By single-peakedness of preferences, we obtain $h^{r+1}\, P_i\, h^r\, P_i\, \cdots\, P_i\, h^{p-1}$, which implies that h^{r+1} is the most preferred element in $\mathrm{Nbr}\big(\Pi^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}, H^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}\big) = \{h^{p-1}, \ldots, h^{r+1}\}$. Hence, $x = h^{r+1}$.

Since h^r and h^{r+1} are adjacent in $H^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}$ and $h^r \in \Pi^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}$, it follows that h^r is the projection of h^{r+1} onto $\Pi^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}$. That is, $x' = h^r$. Therefore, by Lemma .1.3, we obtain $m(\sigma_i(P_i), \sigma_{-i}(P_{-i}))\, R_i\, h^r$.

■

We now complete the proof by considering the two cases: $h' \in \widehat{\Pi}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}$ and $h' = h^{p-1}$.

First, suppose that $h' \in \widehat{\Pi}^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}$. By the definition of the extensive game form $\mathbb{E}$, agent i is then matched with house h'. From the proof of Claim .3.2, we have $h^r\, P_i\, h^{r-1}\, P_i\, \cdots\, P_i\, h^{p-1}$, which implies that $h^r\, R_i\, h'$. Combining this with Claim .3.2, we obtain $m(\sigma_i(P_i), \sigma_{-i}(P_{-i}))\, R_i\, h'$.

Next, suppose that $h' = h^{p-1}$. By side-consistency, along any path from the root node through $v^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}$ to any subsequent node at which agent i takes an action, the agent can only select houses that lie weakly to the left of their current endowment. Consequently, the endowment of agent i moves weakly to the left of h^q, and hence the final allocation must be some house h'' satisfying $h'' \preceq h^q$. Since $h^r \preceq \tau\big(P_i, H^{(\underset{\sim}{i}^1,\ldots,\underset{\sim}{i}^k)}_{(s_1,\ldots,s_{k-1},s)}\big)$, it follows that $h^q\, R_i\, h''$, and because $h^r\, R_i\, h^q$, by Claim .3.2 we conclude that $m(\sigma_i(P_i), \sigma_{-i}(P_{-i}))\, R_i\, h''$.

.4 Illustration of TN Algorithm outcome in Example 6.5.1

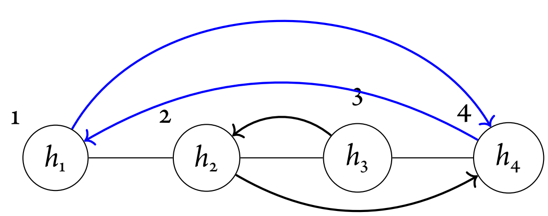

Figure .4.1: Stage 1 of TN Algorithm

Table .4.1: Stage 1: Pointer Sequence and Cycle Formation

Step s	Pointer $\pi(1, s)$	$\Pi(1, s)$	Pointer $\pi(1, s+1)$
1	$\tau(\prec, H_1) = h_3$	$\{h_3\}$	Owner of $\tau(P_3, \{h_2, h_3, h_4\}) = h_2$
2	h_2	$\{h_3, h_2\}$	Owner of $\tau(P_2, \{h_1, h_2, h_3, h_4\}) = h_4$
3	h_4	$\{h_3, h_2, h_4\}$	Owner of $\tau(P_4, \{h_1, h_2, h_3, h_4\}) = h_1$
4	h_1	$\{h_3, h_2, h_4, h_1\}$	$\tau(P_1, H_1) = h_4 \in \Pi(1, 4)$: Cycle found

Cycle: $h_1 \to h_4 \to h_1$

New Endowment e_2**:** $1 \mapsto h_4,\ 4 \mapsto h_1$

Update: Remove agents 1 and 4 with houses h_4 and h_1, respectively.

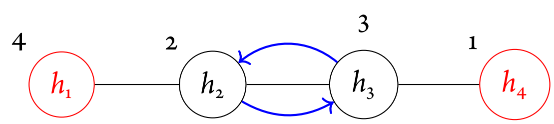

Figure .4.2: Stage 2 of TN Algorithm

Table .4.2: Stage 2: Pointer Sequence and Cycle Formation

Step s	Pointer $\pi(2, s)$	$\Pi(2, s)$	Pointer $\pi(2, s+1)$
1	$\tau(\prec, H_2) = h_3$	$\{h_3\}$	Owner of $\tau(P_3, \{h_2, h_3\}) = h_2$
2	h_2	$\{h_3, h_2\}$	Owner of $\tau(P_2, \{h_2, h_3\}) = h_3 \in \Pi(2, 2)$: Cycle found

Cycle: $h_2 \to h_3 \to h_2$

New Endowment e_3**:** $2 \mapsto h_3,\ 3 \mapsto h_2$

Update: Remove agents 2 and 3 with houses h_3 and h_2, respectively.

.5 Illustration of TN Algorithm outcome in Example 6.5.5

.5.1 Outcome at P_N

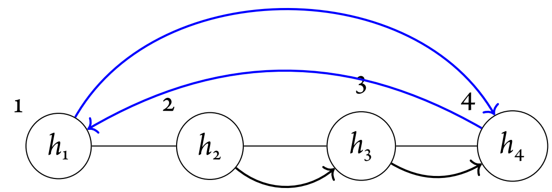

Figure .5.1: Stage 1 of TN Algorithm at P_N

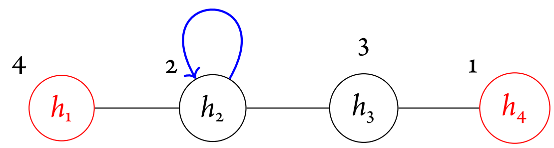

Figure .5.2: Stage 2 of TN Algorithm at P_N

.5.2 Outcome at P'_N

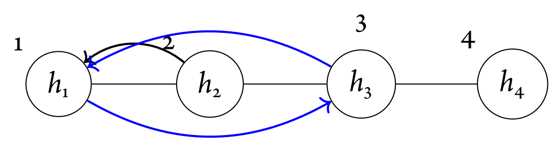

Figure .5.3: Stage 1 of TN Algorithm at P'_N

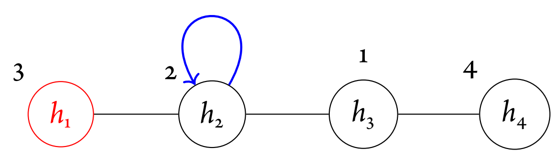

Figure .5.4: Stage 2 of TN Algorithm at P'_N

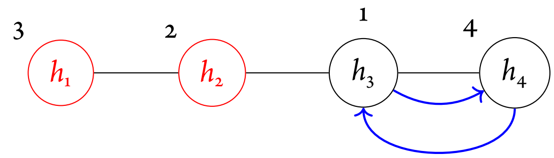

Figure .5.5: Stage 3 of TN Algorithm at P'_N

7

Matching rules and obvious strategy-proofness

7.1 Introduction

7.1.1 Description and background of the problem

This paper investigates matching rules that satisfy individual rationality, Pareto optimality, and obvious strategy-proofness (osp) in the context of one-to-one, one-sided matching problems. While individual rationality and Pareto optimality are well-established concepts in matching theory, obvious strategy-proofness is a relatively recent addition to the literature. Intuitively, osp captures the idea that the matching mechanism is transparent and comprehensible to agents with bounded cognitive capacities.

Formally, a matching rule is said to be obviously strategy-proof if it can be implemented through a mechanism in which truthful reporting constitutes an obviously dominant strategy. This concept is defined in the context of a game form represented by an extensive-form game tree, where each decision node is assigned to an agent. A strategy is obviously dominant if, at every decision node, the worst possible outcome that can result from following the prescribed action is at least as good as the best possible outcome obtainable from deviating at that point. Consequently, even if an agent cannot compute the exact final outcome induced by their strategy, they can still be confident that no deviation will yield a better result, thereby eliminating the incentive for strategic misrepresentation.

7.1.2 Contribution of the Paper

This paper makes two primary contributions to the study of matching mechanisms that satisfy IR, PO, and osp in one-to-one, one-sided matching settings.

General Matching Rules. First, we consider *arbitrary matching rules*. Within this general framework, we introduce a novel condition, termed *bottom freeness*, which we show to be *sufficient* for the non-existence of matching rules that simultaneously satisfy IR, PO, and osp. The bottom freeness condition requires that the domain of preferences contains at least two preference orderings, say P and P', such that the most-preferred alternatives (i.e., the tops) differ, while the least-preferred alternatives (i.e., the bottoms) are identical. This structural property of the preference domain is crucial in identifying the limitations of osp-implementable mechanisms.

As applications of this necessary condition, we derive several *impossibility results*. Specifically, we demonstrate that no matching rule can satisfy IR, PO, and osp on the following domains:

- the *circular domain*, where preferences are induced from a circular ordering over the alternatives,
- the *union of single-peaked and single-dipped domains*, which combines single-peaked and single-dipped preferences together, and
- the *unrestricted domain*, which places no constraints on the agents' preferences.

Top Trading Cycles Rule. Second, we analyze a well-studied class of mechanisms, namely the *Top Trading Cycles* (TTC) rule, and investigate the conditions under which TTC can be consistent with IR, PO, and osp. The TTC rule is one of the most prominent mechanisms in allocation problems without monetary transfers, especially in housing and school choice settings, due to its desirable efficiency and strategy-proofness properties under various domain restrictions.We begin by examining the performance of TTC within *regular single-peaked domains*, a common restriction in social choice theory under which, for every house, the preference domain contains at least one preference that ranks that house at the top. Contrary to what might be expected, we show that the TTC rule *fails to satisfy osp* even within these structured domains. This finding sharpens and extends a result by [2], who had previously established the incompatibility of TTC with osp on the *full single-peaked domain*. Our result demonstrates that the failure of osp persists even under more restrictive assumptions.

To address this negative result, we introduce a novel structural condition on preference domains, which we term the *top two consensus property*. Informally, this property requires that a fixed object (or house) appears among the top two ranked positions in all preference orderings within the domain. We show that when a single-peaked domain satisfies the top-two consensus property, the TTC rule is *osp-implementable*.

Taken together, our results delineate the boundary between possibility and impossibility for osp mechanisms in matching theory and provide both conceptual and technical insights into the design of simple, robust, and efficient allocation rules.

The following table gives a brief summary of the results of this chapter.

Table 7.1.1: Summary of Main Results

Setting	**Domain / Condition**	**Result**
General Matching Rules		
Arbitrary matching rules	Bottom free domains	No rule satisfies IR, PO, and OSP (Impossibility)
Top Trading Cycles (TTC)		
TTC	Regular single-peaked domains	TTC fails OSP (Negative result)
TTC	Single-peaked domains satisfying top-two consensus	TTC is OSP-implementable (Possibility)

7.1.3 Related literature

[28] shows that in the housing market, the competitive outcome always exists (TTC rule), which is in the core of the housing market. [24] shows that when preferences are strict, the TTC outcome is the unique competitive outcome. [23] shows that the TTC rule is strategy proof. [15] shows that the TTC rule is the unique rule in the housing market satisfying the three desirable properties of IR, PO, and SP under unrestricted domains. Thus [15] provides a non cooperative characterization of the TTC rule. However, [2] shows that this uniqueness of the TTC rule does not hold when the domain of preference is restricted to a single-peaked domain. She provides a new, simple algorithm called the crawler that satisfies IR,PO, and SP. Moreover, the Crawler satisfies an even stronger notion of incentive compatibility called obvious strategy-proofness (osp). [13] introduced the notion of osp, which requires a notion of strategy proofness that even cognitively limited agents can understand and recognize as strategy proof.

Recently, many papers in the literature have looked into the problem of characterizing the set of matching rules satisfying osp with other desirable properties like IR, PO, symmetricity (The Random Priority Mechanism is Uniquely Simple, Efficient, and Fair), equal treatment of equals (Strategy-proof, Efficient, and Fair Allocation: Beyond Random Priority), etc. [20] shows that random priority is the only rule that satisfies PO, obvious strategy-proofness, and symmetricity. [3] provides a simplified structure of the extensive game form called the obviously incentive compatible gradual revelation mechanism, which can implement every obviously strategy-proof matching rule. [32] identifies an acyclicity condition that is both necessary and sufficient for a TTC rule to be osp. However, their framework differs significantly from our paper, as the acyclicity condition provided in that paper assumes a priority structure of the houses over the agents. On the other hand, we look at the problem from the lens of a pure one-sided matching problem and provide conditions on the preference domain of the agents. In this paper, we look at the problem in two different ways; In the first part, we explore general matching rules, while in the second part , we ask the same questions with respect to one special matching rule called the TTC rule.

7.2 Preliminaries

Let $N = \{1, \ldots, n\}$ denote a set of n agents and $H = \{h_1, \ldots, h_n\}$ denote a set of n houses. Each agent $i \in N$ has a (strict) preference P_i over the houses in H.[1] We denote the weak part of a preference P with R.[2] For $1 \leq k \leq n$ and a preference P, $r_k(P)$ denotes the k-th ranked house in a preference P, that is, $r_k(P) = a$ if $|\{b \mid bRa\}| = k$. Given a preference P and a subset of houses $\bar{H} \subseteq H$, we denote by $\tau(P, \bar{H})$ the top-ranked house in $\bar{H}$ according to P; that is, $\tau(P, \bar{H})\, R\, h$ for all $h \in \bar{H}$. Given a set of preferences $\mathcal{D}$ and a subset of houses $\bar{H} \subseteq H$, $\tau(\mathcal{D}, \bar{H})$ denotes the set of houses that are top-ranked in $\bar{H}$ according to P for some preference P in $\mathcal{D}$, that is, $\tau(\mathcal{D}, \bar{H}) = \cup_{P \in \mathcal{D}}(\tau(P, \bar{H})$. A collection of preferences, one for each agent, is called a preference profile. We denote a preference profile $(P_1, \ldots, P_n)$ by P_N. The set of all admissible preferences for an agent i is denoted by $\mathcal{P}_i$ and that of all admissible preference profiles is denoted by $\mathcal{P}_N$.

A bijection $\mu : N \rightarrow H$ is called a matching, where $\mu(i) = h$ implies that agent i is matched to house h under μ. Let $\mathcal{M}$ be the set of all matchings.

A matching rule is a mapping $\phi : \mathcal{P}_N \rightarrow \mathcal{M}$ that assigns a matching to each preference profile in $\mathcal{P}_N$.

The houses are initially matched with the agents according to a matching e, which is called the initial endowment. Without loss of generality, we assume $e(i) = h_i$ for all $i \in N$. We use the notation $o(h)$ to denote the owner of the house h. Thus $o(h) = i$ if and only if $e(i) = h$.

We define some desirable properties for a matching rule. A matching μ is individually rational if each agent receives a house that is at least as good as their endowed house. A matching rule is individually rational if it assigns an individually rational matching at every profile.

Definition 7.2.1 *A matching $\mu \in \mathcal{M}$ is **individually rational (IR)** at a preference profile $P_N \in \mathcal{P}_N$ if for each $i \in N$, $\mu(i) R_i e(i)$. A matching rule $\phi : \mathcal{P}_N \rightarrow \mathcal{M}$ is individually rational (IR) if $\phi(P_N)$ is IR at P_N for every $P_N \in \mathcal{P}_N$.*

A matching μ is said to be Pareto optimal at a preference profile if there is no other matching where each agent is weakly better off and at least one agent is strictly better off. A matching rule is Pareto optimal if it assigns a Pareto optimal matching at every preference profile.

[1] A strict preference on a set S is a complete, transitive, and anti-symmetric binary relation on S.

[2] Since P is a strict preference, aRb implies either aPb or $a = b$.

Definition 7.2.2 *A matching $\mu \in \mathcal{M}$ is* ***Pareto optimal (PO)*** *at a preference profile $P_N \in \mathcal{P}_N$ if there exists no other matching $\mu' \in \mathcal{M} \setminus \mu$ such that for each $i \in N$, $\mu'(i) R_i \mu(i)$. A matching rule $\phi : \mathcal{P}_N \to \mathcal{M}$ is Pareto optimal (PO) if $\phi(P_N)$ is PO at every $P_N \in \mathcal{P}_N$.*

A matching rule is strategy-proof if no agent can be better off by misreporting their true preference.

Definition 7.2.3 *A matching rule $\phi : \mathcal{P}_N \to \mathcal{M}$ is* ***strategy-proof*** *if for all $i \in N$, all $P_i, P'_i \in \mathcal{P}_i$, and all $P_{-i} \in \mathcal{P}_{-i}$, we have $\phi_i(P_i, P_{-i}) R_i \phi_i(P'_i, P_{-i})$.*

Strategy-proofness ensures that no agent can misreport their true preferences and get a better outcome. Although strategy-proofness ensures that truth-telling is a weakly dominant strategy for each agent, agents may not always understand the matching rule well enough to realize that. [13] introduces the notion of obvious strategy-proofness to recognize rules which not only are strategy-proof, but can be identified to be strategy-proof by cognitively limited agents.

Definition 7.2.4 *An* ***extensive game form*** *is defined as the tuple $\mathbb{E} = \langle T, N, \gamma \rangle$ where*

- *$T = (V, E)$ is a directed rooted tree with the root denoted by v_0 and the set of terminal nodes denoted by Z,*
- *N denotes the set of players, and*
- *$\gamma : V \setminus Z \to N$ denotes the player assigning function that assigns a player in N at each non-terminal node.*

The outgoing edges from a node v are denoted by $O(v)$. For $i \in N$, let V_i denote the nodes of V where i is assigned, that is, $V_i = \{v \mid \gamma(v) = i\}$.

Given an extensive game form $\mathbb{E}$, an action of agent $i \in N$ is a mapping $a_i : V_i \to E$ such that $a_i(v_i) \in O(v_i)$ for all $v_i \in V_i$. In other words, an action of agent i assigns an outgoing edge from every node in V_i. We denote by A_i the set of all actions of agent i, and by A_N the set of all action profiles $\prod_{i \in N} A_i$ of all agents.

A strategy s_i is defined as a mapping $s_i : \mathcal{P}_i \to A_i$. In other words, a strategy of agent i assigns an action of that agent to each preference of agent i. We denote by S_i the set of all strategies of agent i. A strategy profile s_N of all agents in N is an element of $S_N := \prod_{i \in N} S_i$. For a preference profile P_N, we use the notation $s_N(P_N)$ to denote the action profile $\prod_{i \in N} s_i(P_i)$.

A mechanism is a mapping $m : A_N \to \mathcal{M}$. Since each action profile uniquely determines a terminal node, a mechanism can be viewed as a function that assigns a matching to every terminal node.We say that a mechanism $m : A_N \to \mathcal{M}$ implements a matching rule $\phi : \mathcal{P}_N \to \mathcal{M}$ if $\phi(P_N) = m(s_N(P_N))$ for all $P_N \in \mathcal{P}_N$.

For $v \in V$, we denote by T_v the restriction of the tree T rooted at v, and by $\mathbb{E}_v$ the restriction of $\mathbb{E}$ on T_v. We call $\mathbb{E}_v$ the sub-game extensive form structure rooted at v. Similarly, we denote by s_i^v and s_N^v the restrictions of s_i and s_N on T^v.

Definition 7.2.5 *A strategy $s_i \in S_i$ of agent i is **obviously dominant** for agent $i \in N$ in a mechanism m, if for each $v_i \in V_i$, each $P_N \in \mathcal{P}_N$, each $\bar{s}_i \in S_i$ such that $s_i(P_i)(v_i) \neq \bar{s}_i(P_i)(v_i)$, and each $s_{-i} \in S_{N \setminus i}$, we have*

$$m(s_N^{v_i}(P_N))R_i m(\bar{s}_i^{v_i}(P_i), s_{-i}^{v_i}(P_{-i})).$$

Definition 7.2.6 *A matching rule $\phi : \mathcal{P}_N \to \mathcal{M}$ is **obviously strategy-proof** if there exists a mechanism m and a strategy profile s_N such that*

(i) s_i is obviously dominant for each agent $i \in N$, and

(ii) m implements ϕ.

7.3 General matching rule

In this section, we look at domains under which every IR and PO matching rule fails to satisfy obvious strategy-proofness.

We first define the notion of bottom-freeness. A domain of preferences said to be bottom-free if for every house, there exists a two preferences in the domain such that the house is the last ranked alternative in this preference and the top ranked alternatives in the two preferences are different.

Definition 7.3.1 *A set of preferences $\mathcal{P}$ satisfies **bottom-freeness** if for every $h \in H$, there are two preferences $P, P' \in \mathcal{P}$ with $r_1(P) \neq r_1(P')$ such that $r_n(P) = r_n(P') = h$.*

Several well-known domains of preferences such as the completely unrestricted domain, the unrestricted domain, the circular domain, and the union of single-peaked and single-dipped domain satisfy bottom-freeness. We are now ready to state the main result of this section.

Theorem 7.3.2 *Suppose the set of admissible preferences $\mathcal{P}_i$ of each agent i satisfies the bottom-freeness property. Then, there is no matching rule $\phi : \mathcal{P}_N \to \mathcal{M}$ satisfying IR, PO, and obvious strategy-proofness.*

Before presenting the proof of the theorem, we introduce a result from [3], along with the associated definitions, which will be used throughout the paper. Theorem 3.1 in [3] states that a matching rule is obviously strategy-proof if and only if some obviously incentive compatible gradual revelation mechanism implements it. A gradual revelation mechanism is a mechanism where the underlying extensive game form contains no singleton action choices at any node and no directly consecutive moves by the same agent. Each non-terminal node $z \in Z$ is associated with a set of preference profiles denoted by $\mathcal{R}_N(v) = \prod_{i \in N} \mathcal{R}_i(v)$ such that $\mathcal{R}_N(v_0) = \mathcal{P}_N$. At each node v with $\gamma(v) = i$, $\{\mathcal{R}_i(v, a) \mid a \in O(v)\}$ partitions $\mathcal{R}_i(v)$ and $\mathcal{R}_j(v) = \mathcal{R}_j(v, a)$ for all $j \neq i \in N$ and all $a \in O(v)$. A strategy s_i for a player $i \in N$ is said to be truthful in a gradual revelation mechanism if $P_i \in \mathcal{R}_i(v_i, s_i(P_i)(v_i))$ for every non-terminal node $v_i \in V_i$ and all $P_i \in \mathcal{P}_N$. A gradual revelation mechanism is said to be obviously incentive compatible if the truthful strategy s_i is obviously dominant for each agent $i \in N$ in the mechanism m. An obviously incentive compatible gradual revelation mechanism m implements a matching rule $\phi : \mathcal{P}_N \to \mathcal{M}$ if $\phi(P_N) = m(s_N(P_N))$ for all $P_N \in \mathcal{P}_N$.

Proof: Suppose that the domain $\mathcal{P}_i$ satisfies bottom-freeness for each $i \in N$. Let us assume for contradiction that there exists a matching rule $\phi : \mathcal{P}_N \to \mathcal{M}$ such that ϕ satisfies IR, PO, and obvious strategy-proofness. We know from Theorem 3.1 in [3], that a matching rule is obviously strategy-proof if and only if some obviously incentive compatible gradual revelation mechanism implements it. In what follows, we show that no obviously incentive compatible gradual revelation mechanism can implement ϕ. We first construct a subset of the preference domain in the following way. Let $\mathcal{D}_i = \{P_i \in \mathcal{P}_i \mid r_1(P_i) \neq h_i\}$. Define $\mathcal{D}_N$ as $\prod_{i \in N} \mathcal{D}_i$.

Claim 7.3.1 *For every node v in M, $\mathcal{D}_N \subseteq \mathcal{R}_N(v)$ implies that there exists an action $a \in O(v)$ with $\mathcal{D}_N \subseteq \mathcal{R}_N(v, a)$.*

Proof: [Proof of Claim 7.3.1] Consider any node v such that $\mathcal{D}_N \subseteq \mathcal{R}_N(v)$. Let $\gamma(v) = i$. Consider the preference $P_i^* \in \mathcal{D}_i$ such that $r_m(P_i^*) = h_i$. Since $\mathcal{P}_i$ satisfies bottom-freeness, such a P_i^* always exists in $\mathcal{P}_i$. Since $r_m(P_i^*) = h_i$, $r_1(P_i^*) \neq h_i$. Therefore $P_i^* \in \mathcal{D}_i$. Since $P_i^* \in \mathcal{D}_i \subseteq \mathcal{R}_i(v)$, there exists an action $a \in O(v)$ such that $P_i^* \in \mathcal{R}_i(v, a)$. Since for each $j \neq i$, $\mathcal{P}_j$ satisfies bottom-freeness, there exists atleast two preferences P_j' and $P_j'' \in \mathcal{P}_j$ such that $r_m(P_j') = h_i$ and $r_m(P_j'') = h_i$ and $r_1(P_j') \neq r_1(P_j'')$. Thus either $r_1(P_j') \neq h_j$ or $r_1(P_j'') \neq h_j$ or both. Therefore, atleast one of P_j' and P_j'' must belong to $\mathcal{D}_j$ and hence for each $j \in N \setminus \{i\}$, there exists a preference $P_j \in \mathcal{D}_j$ such that $r_m(P_j) = h_i$. Since ϕ satisfies IR, $\phi_j(P_i^*, P_{-i}) \neq h_i$ for all $j \neq i$. Therefore we must have $\phi_i(P_i^*, P_{-i}) = h_i$. Now consider any other preference $\bar{P}_i \in \mathcal{D}_i$. Without loss of generality, assume that $r_1(\bar{P}_i) = h_k$. For each agent $j \in N \setminus \{i\}$, consider the preference $\bar{P}_j$ such that $r_m(\bar{P}_j) = h_k$. Using similar logic as earlier, we can argue that $\bar{P}_j$ exists in $\mathcal{D}_j$ for all $j \in N \setminus \{i\}$. Since ϕ satisfies IR, $\phi_j(\bar{P}_i, \bar{P}_{-i}) \neq h_k$ for all $j \neq \{i, k\}$. Using Pareto optimality, we have $\phi_i(\bar{P}_i, \bar{P}_{-i}) = h_k$. Hence we have $\phi_i(\bar{P}_i, \bar{P}_{-i}) P_i^* \phi_i(P_i^*, P_{-i})$, which in turn implies that the action a under truthful strategy at preference P_i^* is not obviously dominant. As we have taken $\bar{P}_i$ to any arbitrary preference in $\mathcal{D}_i$ other than P_i^*, it follows that the set of preferences $\mathcal{D}_i$ cannot be partitioned through the outgoing edges at the node v, that is, $\mathcal{D}_i \subseteq \mathcal{R}_i(v, a)$. This, along with the fact that $\gamma(v) = i$ was arbitrary, gives us that $\mathcal{D}_N \subseteq \mathcal{R}_N(v, a)$. This completes the proof of the claim. ■

By definition, $\mathcal{D}_N \subseteq \mathcal{R}_N(v_0)$. By Claim 7.3.1, there exists an action $a \in O(v_0)$ such that $\mathcal{D}_N \subseteq \mathcal{R}_N(v_0, a)$. Continuing in this manner, we have $\mathcal{D}_N \subseteq \mathcal{R}_N(z)$ for some terminal node $z \in Z$. We know that there exist two different preference profiles in $\mathcal{D}_N$, namely (P_i, P_{-i}) and $(\bar{P}_i, \bar{P}_{-i})$ mentioned earlier such that $\phi(P_i, P_{-i}) \neq \phi(\bar{P}_i, P'_{-i})$. This along with the fact that (P_i, P_{-i}) and $(\bar{P}_i, \bar{P}_{-i}) \in \mathcal{R}_N(z)$ for some $z \in Z$, we have a contradiction to the obvious strategy proof property of ϕ. This completes the proof of the theorem. ■

7.4 TTC rule

In this section, we narrow our focus from the set of all individually rational, Pareto optimal, and obviously strategy-proof matching rules to a specific rule known as the Top Trading Cycle (TTC) rule, originally introduced by Shapley and Scarf [28]. The TTC rule is well-established in the literature and has broad applications—not only in one-sided matching environments such as the housing market, but also in two-sided settings like the marriage market (see, for example, generalizations of TTC introduced in [21], [19] [1], [26]).

It is known that the TTC rule satisfies individual rationality and Pareto optimality on the domain of unrestricted preferences [23], and therefore, it retains these properties on any restricted domain. Li [13] show that under unrestricted preferences, and with three or more agents, the TTC rule fails to satisfy obvious strategy-proofness. However, Bade [2] prove in Theorem 5 that the TTC rule is obviously strategy-proof on the domain of single-peaked preferences $\mathcal{S}$ if and only if there are at most three agents. Proposition 7.4.1 in this section slightly strengthens this result by showing that the failure of TTC to satisfy osp does not require the full richness of the maximal single-peaked domain. Finally theorem 7.4.6, provides a subset of the single-peaked domain where the TTC rule is obviously strategy-proof.

A prior order $\prec$ on the set H is a strict ordering of the elements of H.[3] A preference P on H is single-peaked with respect to a prior order $\prec$ on H if preference declines as one goes farther away (with respect to $\prec$) from their top-ranked (peak) house.

Definition 7.4.1 *A preference P on H satisfies single-peakedness with respect to a prior order $\prec$ on H if for all $a, b \in H$, aPb whenever $[r_1(P) \prec a \prec b$ or $b \prec a \prec r_1(P)]$.*

We denote by $\mathcal{S}$ the set of all single-peaked preferences on H with respect to $\prec$[4].

Definition 7.4.2 *A set of single-peaked preferences $\mathcal{S}^R \subseteq \mathcal{S}$ satisfies **regularity** if for every $h \in H$, there exists a preference $P \in \mathcal{S}^R$ such that $r_1(P) = h$.*

Our next proposition shows that even under mild regularity conditions on the single-peaked domain, the Top Trading Cycles rule fails to satisfy the property of obvious strategy-proofness.

Proposition 7.4.1 *If the set of admissible single-peaked preferences $\mathcal{S}_i$ of each agent i satisfies regularity, then the matching rule $\phi^T : \mathcal{S}^R_N \to \mathcal{M}$, given by the TTC rule is not obviously strategy-proof.*

The proof of this proposition is similar to the proof of Theorem 5 in [2]. We provide a brief example to illustrate the working of the proof.

[3] Technically, a prior order is a preference. We do not call it a preference as it does not represent the same.

[4] Throughout the paper, we assume a fixed $\prec$ whenever we discuss about single-peaked domain.

Example 7.4.3 *Consider $N = \{1, 2, 3, 4\}$ where agent i is endowed with h_i. Let the preference domain be defined in the following manner.*

$$\mathcal{S}_1^R = \{P^1, P^2, P^3, P^4\},$$
$$\mathcal{S}_2^R = \{P^1, P^2, P^3, \bar{P}^3, P^4\},$$
$$\mathcal{S}_3^R = \{P^1, P^2, \bar{P}^2, P^3, \bar{P}^3, P^4\},$$
$$\mathcal{S}_4^R = \{P^1, P^2, \widehat{P}^3, \bar{P}^3, P^4\}.$$

where

Table 7.4.1: Preferences

P^1	P^2	$\bar{P}^2$	P^3	$\bar{P}^3$	$\widehat{P}^3$	P^4
h_1	h_2	h_2	h_3	h_3	h_3	h_4
h_2	h_1	h_3	h_2	h_4	h_2	h_3
h_3	h_3	h_4	h_1	h_2	h_4	h_2
h_4	h_4	h_1	h_4	h_1	h_1	h_1

We show the TTC rule $\phi^T : \mathcal{S}_N^R \to \mathcal{M}$ to be not osp using the same arguments as in the proof of Theorem 7.3.2. We construct a permissible subset of the preference domain, denoted by $\prod_i \mathcal{D}_i$in the following manner.

$$\mathcal{D}_1 = \{P^2, P^3, P^4\},$$
$$\mathcal{D}_2 = \{P^3, \bar{P}^3, P^4\},$$
$$\mathcal{D}_3 = \{P^1, P^2, \bar{P}^2\},$$
$$\mathcal{D}_4 = \{P^1, P^2, \widehat{P}^3, \bar{P}^3\}.$$

We show that Claim 7.3.1 holds true for $\mathcal{D}_N$. We provide the logic for agents 1 and 2; the arguments for agents 3 and 4 follow similarly. Consider agent 1 and the preference $P_1 = P^4$. Note that we can always choose the preferences of the other agents, P_{-1}, such that they form a TTC cycle among themselves(such as, $P_2 = P^4$, $P_3 = P^2$, and $P_4 = \widehat{P}^3$), which implies that agent 1 receives h_1 at (P_1, P_{-1}). Consider the preference $P_1' = P^2$. If we take $P_2' = P^3$ and $P_3' = P^1$, then agents 1,2, and 3 form a TTC cycle and agent 1 receives h_2 which is preferred to h_1 under P_1.Consider the preference $P_1' = P^3$. If we take $P_3' = P^1$, then agents 1 and 3 form a TTC cycle and agent 1 receives h_3 which is preferred to h_1 under P_1. This completes all cases for agent 1 in the proof of Claim

7.3.1.

For agent 2, consider the preference $P_2 = P^4$. At P_{-2} such that $P_1 = P^4$, $P_3 = P^1$, and $P_4 = \bar{P}^3$, all agents other than 2 form a TTC cycle and agent 2 receives h_2 at (P_2, P_{-2}). Consider the preference $P_2' = P_3$ (or $P_2' = \bar{P}_3$). If we take $P_3' = P^2$, then agents 2 and 3 form a TTC cycle and agent 2 receives h_3 which is preferred to h_2 under P_2. This completes all cases for agent 2 in the proof of Claim 7.3.1.

Next we state a sufficient condition on a set of single-peaked preferences so that the TTC rule is obviously strategy-proof. A set of single-peaked preferences is said to satisfy top-two consensus property, if there exists at least one house h such that, h is either the first or the second ranked alternative for each preference in the set.

Definition 7.4.4 *A set of single-peaked preferences $\bar{\mathcal{S}} \subset \mathcal{S}$ satisfies top-two consensus property if there exists a house $h^* \in H$ such that $h^* \in \{r_1(P), r_2(P)\}$ for each $P \in \bar{\mathcal{S}}$.*

Remark 7.4.5 *If a set $\bar{\mathcal{S}}$ satisfies the top-two consensus property, then the set of top ranked house i.e., $\{r_1(P) \mid P \in \bar{\mathcal{S}}\}$ can contain at most three houses namely h_j, h_{j-1}, and h_{j+1} where h_{j-1} and h_{j+1} are the immediate neighbors of the house h_j on its two sides with respect to the linear order $\prec$. That is, $h_{j-1} \prec h \prec h_{j+1}$, and there does not exist any house $\bar{h}, \hat{h} \in H \setminus \{h_j, h_{j-1}, h_{j+1}\}$ such that $h_{j-1} \prec \bar{h} \prec h_j \prec \hat{h} \prec h_{j+1}$.*

We now present the main theorem of this section, which states that the TTC rule satisfies *obvious strategy-proofness* on a single-peaked domain, provided the domain satisfies the *top-two consensus* property. It is important to note that all agents are assumed to have the same domain of admissible preferences.

Theorem 7.4.6 *If a set of single-peaked preferences $\bar{\mathcal{S}}$ satisfies the top-two consensus property, then the matching rule $\phi^T : \bar{\mathcal{S}}^n \to \mathcal{M}$, given by the TTC rule, is obviously strategy-proof.*

Proof: Before beginning with the proof, let us introduce some notations that we require in the proof. Given a directed tree $T = (V, E)$, the path from node v to v' is defined as a sequence of distinct nodes $(v^1, v^2, \ldots, v^k)$ such that $(v^i, v^{i+1}) \in E$ for all $i = 1, 2, \ldots, k-1$ and $v^1 = v$ and $v^k = v'$. Since T is a tree, there exists a unique path from any node v to any other node v'. We denote the unique path by $[v, v']$. Two nodes v, v' are said to be adjacent nodes if $(v, v') \in E$. For any two adjacent nodes v, v', we use the notation $L((v, v'))$ to denote the label of the edge connecting the two nodes. We define a linear order $<$ on the set of houses in the following manner: $h_j < h_{j-1} < h_{j-2} < \cdots < h_1 < h_{j+1} < h_{j+2} < \cdots < h_n$.

To prove Theorem 7.4.6, we first construct an extensive game form for the implementation of TTC matching rule. Next we define a strategy profile in that game form and argue that it implements the TTC rule. Finally we show that the strategy of each player is obviously dominant.

Recall the definition of an extensive game form in Definition 7.2.4. We define an extensive game form $\mathbb{E} = \langle T, N, \gamma \rangle$, where T is a tree, V is the set of nodes, Z is the set of terminal nodes, and γ is the player assigning function, in the following manner. We label each non-terminal node $v \in V \setminus Z$ with a vector $(N(v), H(v), \Pi(v))$ where $N(v)$ denotes the set of (remaining) agents at node v, $H(v)$ denotes the set of (remaining) houses at node v, and the set $\Pi(v)$ is a subset of $H(v)$. Given the labels of every edge on the path from the root v_0 to a node v, we say that a house $h \in O(v)$ is side consistent at v with player assignment $\gamma(v) = i$, for some $i \in N(v)$, if the following conditions hold:

- If $h \prec h_i$, then for every node v_u on the path $[v_1, \ldots, v_t]$ from the root $v_1 = v_0$ to $v_t = v$ such that $\gamma(v_u) = i$, it holds that
$$L\big((v_u, v_{u+1})\big) \prec h_i.$$
- If $h_i \prec h$, then for every such node v_u, it holds that
$$h_i \prec L\big((v_u, v_{u+1})\big).$$

Observe that if there is no node on the path $[v_1, \ldots, v_t]$ with the same player assignment as node v, then every house at node v is side consistent. Given a set of houses $O(v)$ at node v, we denote the subset of side consistent houses in $O(v)$ with $\widehat{O}(v)$.

For each non-terminal node v, we have $|N(v)| > 1$, and for each terminal node v, we have $|N(v)| = 1$. At each non-terminal node v, the number of edges emanating from v is equal to $|\widehat{O}(v)|$ where $|O(v)| = |\tau(\bar{S}, H(v))|$. In other words, each outgoing edge is uniquely labeled by a side consistent house that appears as a top-ranked house in the set of preferences $\bar{S}$ restricted to the set of houses $H(v)$ that remain available at the node v.

Fix the label at the root node v_0 as $(N(v_0), H(v_0), \Pi(v_0)) = (N, H, h_j)$ and let $\gamma(v_0) = j$. Recall that agent j is the owner of the house h_j. Consider any non terminal node v' that is adjacent to v through the edge labeled as h, which is $L((v, v')) = h$.

1. If $h \notin \Pi(v)$, then the label of v' is updated as follows: $N(v') = N(v), H(v') = H(v)$, $\Pi(v') = \Pi(v) \cup \{h\}$. And $\gamma(v') = o(h)$.

2. If $h \in \Pi(v)$, then the label of v' is updated as follows: First, we trace the path $[v_0, v']$ from the root v_0 to the node v'. We identify the node, say $\bar{v}$, that is closest to v' along this path with the property that $\gamma(\bar{v}) = o(h)$. Such a node will exist because $h \in \Pi(v)$ and $\Pi(v)$ is the collection of houses of the agents assigned to play at each subsequent node in a path. Denote the path $[\bar{v}, v']$ as $[v_1, \ldots, v_r]$ where $v_1 = \bar{v}$ and $v_r = v'$. The updated labels at v' are $N(v') = N(v) \setminus \{\gamma(v_1), \ldots, \gamma(v_r)\}$, $H(v') = H(v) \setminus \{e(\gamma(v_1)), \ldots, e(\gamma(v_r))\}$, $\Pi(v') = \tau(<, H(v'))$, and $\gamma(v') = o\big(\tau(<, H(v'))\big)$.

For each $i = 1, \ldots, r$, we match the agent $\gamma(v_i)$ with the house $e(\gamma(v_{i+1}))$ (where $v_{r+1} = v_0$) and remove them from the extensive game form.

Remark 7.4.7 *At any node v, the set $\Pi(v)$ represents the set of houses that the agent $\gamma(v)$ can obtain and leave the extensive game form.*

We illustrate the first few nodes of the extensive game form in the Figure 7.4.1.

For every agent $i \in N$, we define the strategy s_i such that for every $P_i \in \bar{\mathcal{S}}$ and at every $v \in V_i$, $s_i(P_i)(v) = \tau(P_i, L(O(v)))$. Thus strategy for each agent dictates them to report the best house among the remaining set of houses according to their true preference at each node where they are assigned to play. Given a preference profile P_N, let $m(s_N(P_N))$ denote the matching outcome associated with the terminal node that is reached by following the path of $s_N(P_N)$.

It is evident that m with the strategy profile s_N implements the TTC rule ϕ^T since whenever agents leave the mechanism, they leave by forming a cycle (denoted by the path $[\bar{v}, v']$ in the above paragraph) and at each node, the player assigned to play always selects the house which is their best ranked house among the set of remaining houses.

Before proving the obvious dominance of the strategy, we first make some observations regarding the structure of the action set $\widehat{O}(v)$ at each node v. For each node v, $h_{\gamma(v)} \in \widehat{O}(v)$. Observe that due to the way

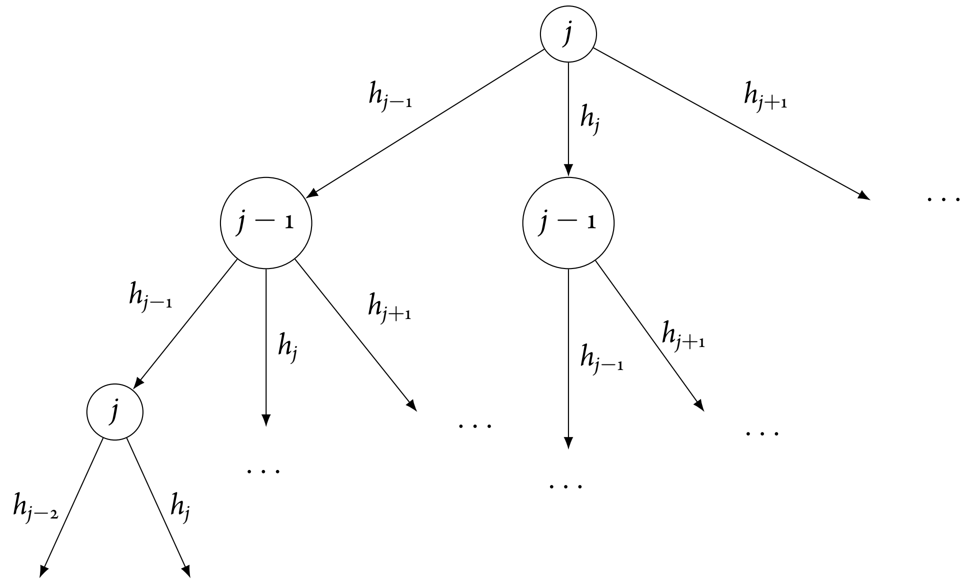

Figure 7.4.1: Illustration of the extensive game form

houses are removed from the extensive game form and the fact that $O(v)$ contains one edge for each house in $\tau(\bar{S}, H(v))$, we have, by Remark 7.4.5, $|\widehat{O}(v)| \leq 3$. In particular, $|\widehat{O}(v)| = 3$ implies that $h_j \in H(v)$.

We now show that s_i is obviously dominant for each agent $i \in N$. Consider any agent $i \in N$ endowed with h_i and any node $v \in V_i$. If $s_i(P_i)(v) \in \Pi(v)$, then they get their best house among the remaining set of houses. Thus, agent i cannot be made strictly better off by any collective deviation starting at v. Consider the case when $s_i(P_i)(v) \notin \Pi(v)$. We distinguish two cases: $|\widehat{O}(v)| = 2$ and $|\widehat{O}(v)| = 3$.

Consider the case when $|\widehat{O}(v)| = 2$. Since $\gamma(v) = i$, we must have $h_i \in \widehat{O}(v)$. From the definition of the extensive game form, we have $h_i \in \Pi(v)$. Combining this with the fact that $s_i(P_i)(v) \notin \Pi(v)$, the only action agent i can take by deviating from their true strategy is to choose the action corresponding to the house h_i. Since $h_i \in \Pi(v)$, agent i gets h_i by deviating from $s_i(P_i)(v)$. However, by individual rationality of the TTC rule, h_i is weakly worse than the truthful outcome.

Suppose $|\widehat{O}(v)| = 3$. Then $h_j \in H(v)$. The top-two consensus property, together with the way houses are matched and removed in the definition of the extensive game form, implies that $\widehat{O}(v)$ consists of h_j and the two immediate neighbors of h_j that belong to $H(v)$. That is,

$$\widehat{O}(v) = \{h_{j-t}, h_j, h_{j+t'}\},$$

where

$$h_{j-t+1}, \ldots, h_{j-1}, h_{j+1}, \ldots, h_{j+t'-1} \notin H(v).$$

Note that the condition $s_i(P_i)(v) \notin \Pi(v)$ implies that it suffices to consider the two cases in which $\Pi(v) \subsetneq \widehat{O}(v)$, namely:

$$\Pi(v) = \{h_j\} \quad \text{or} \quad \Pi(v) = \{h_j, h_i\} \text{ where } h_i \in \{h_{j-t}, h_{j+t'}\} \text{ and } \gamma(v) = o(h_i).$$

Consider first the case $\Pi(v) = \{h_j\}$. Then $\gamma(v) = j$, and agent j truthfully chooses one of h_{j-t} or $h_{j+t'}$. Without loss of generality, suppose agent j truthfully chooses h_{j-t}. Side consistency implies that at any subsequent node v' along the path following from v with $\gamma(v') = j$, we have
$h \notin \widehat{O}(v')$ for every $h \in H(v')$ such that $h_j \prec h$. Therefore, when agent j chooses h_{j-t} at node v, the agent eventually receives a house h such that $h \preceq h_j$. By similar reasoning, when agent j chooses $h_{j+t'}$ at node v, the agent eventually receives a house h such that $h_j \preceq h$. Single-peakedness of preferences, together with the fact that agent j receives h_j by choosing the action corresponding to h_j and the individual rationality of the TTC rule, implies that agent j cannot obtain a better outcome by deviating from $s_j(P_j)(v)$ at node v.

Consider the remaining case in which $\Pi(v) = \{h_j, h_i\}$, where $h_i \in \{h_{j-t}, h_{j+t'}\}$ and $\gamma(v) = o(h_i)$. Without loss of generality, suppose that $h_i = h_{j-t}$. Since $s_i(P_i)(v) \notin \Pi(v)$, agent i must truthfully choose $h_{j+t'}$ under P_i at node v. By the definition of the extensive game form, at any subsequent node v' along the path following from v with $\gamma(v') = i$, we have $h_{j-t}, h_j \in \Pi(v')$. Thus, the worst outcome agent i can obtain by truthfully following strategy $s_i(P_i)(v)$ at node v coincides with the best outcome the agent can obtain by deviating from $s_i(P_i)(v)$ at node v. This completes the proof of the theorem.

■

References

[1] Stergios Athanassoglou and Jay Sethuraman. House allocation with fractional endowments. *International Journal of Game Theory*, 40(3):481–513, 2011.

[2] Sophie Bade. with single-peaked preferences. *Journal of Economic Theory*, 180:81–99, 2019. ISSN 0022-0531. doi: https://doi.org/10.1016/j.jet.2018.12.004. URL https://www.sciencedirect.com/science/article/pii/S0022053118306987.

[3] Sophie Bade and Yannai A. Gonczarowski. Gibbard-satterthwaite success stories and obvious strategyproofness. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, EC '17, page 565, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450345279. doi: 10.1145/3033274.3085104. URL https://doi.org/10.1145/3033274.3085104.

[4] Venkatesh Bala and Sanjeev Goyal. A noncooperative model of network formation. *Econometrica*, 68(5):1181–1229, 2000. doi: https://doi.org/10.1111/1468-0262.00155. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/1468-0262.00155.

[5] Mohit Bhargava, Dipjyoti Majumdar, and Arunava Sen. Incentive-compatible voting rules with positively correlated beliefs. *Theoretical Economics*, 10(3):867–885, 2015.

[6] Francis Bloch and Matthew Olckers. Friend-based ranking in practice. *AEA Papers and Proceedings*, 111:pp. 567–571, 2021. ISSN 25740768, 25740776. URL https://www.jstor.org/stable/27042370.

[7] Gabrielle Demange. Single-peaked orders on a tree. *Mathematical Social Sciences*, 3(4):389–396, 1982.

[8] Bhaskar Dutta and Arunava Sen. Implementation under strong equilibrium: A complete characterization. *Journal of Mathematical Economics*, 20(1):49–67, 1991. ISSN 0304-4068. doi:

https://doi.org/10.1016/0304-4068(91)90017-N. URL
https://www.sciencedirect.com/science/article/pii/030440689190017N.

[9] Zhen Huang and Guoqiang Tian. General exchange rules with single-peaked preferences. March 2023. URL https://people.tamu.edu/~gtian/General_exchange_rules.pdf.

[10] Matthew O. Jackson. *Social and Economic Networks*. Princeton University Press, Princeton, NJ, 2008. ISBN 9780691134406, 0691134405. doi: 10.1515/9781400833993.

[11] Matthew O. Jackson and Asher Wolinsky. A strategic model of social and economic networks. *Journal of Economic Theory*, 71(1):44–74, 1996. ISSN 0022-0531. doi: https://doi.org/10.1006/jeth.1996.0108. URL https://www.sciencedirect.com/science/article/pii/S0022053196901088.

[12] Madhuparna Karmokar and Souvik Roy. On the equivalence of ordinal bayesian incentive compatibility and dominant strategy incentive compatibility for random rules. 2018.

[13] Shengwu Li. Obviously strategy-proof mechanisms. *American Economic Review*, 107(11):3257–87, November 2017. doi: 10.1257/aer.20160425. URL https://www.aeaweb.org/articles?id=10.1257/aer.20160425.

[14] Peng Liu. Neighborhood top trading cycles. 2022. URL https://ssrn.com/abstract=4310598. SSRN working paper.

[15] Jinpeng Ma. Strategy-Proofness and the Strict Core in a Market with Indivisibilities. *International Journal of Game Theory*, 23(1):75–83, 1994. URL https://ideas.repec.org/a/spr/jogath/v23y1994i1p75-83.html.

[16] Dipjyoti Majumdar and Arunava Sen. Ordinally bayesian incentive compatible voting rules. *Econometrica*, 72(2):523–540, 2004.

[17] Debasis Mishra. Ordinal bayesian incentive compatibility in restricted domains. *Journal of Economic Theory*, 163:925–954, 2016.

[18] Roger B. Myerson. Graphs and cooperation in games. *Mathematics of Operations Research*, 2(3): 225–229, 1977. ISSN 0364765X, 15265471. URL http://www.jstor.org/stable/3689511.

[19] Szilvia Papai. Strategyproof assignment by hierarchical exchange. *Econometrica*, 68(6):1403–1433, 2000.

[20] Marek Pycia and Peter Troyan. A theory of simplicity in games and mechanism design. *Econometrica*, 91(4):1495–1526, 2023. doi: https://doi.org/10.3982/ECTA16310. URL https://onlinelibrary.wiley.com/doi/abs/10.3982/ECTA16310.

[21] Szilvia Pápai. Exchange in a general market with indivisible goods. *Journal of Economic Theory*, 132 (1):208–235, 2007. ISSN 0022-0531. doi: https://doi.org/10.1016/j.jet.2005.05.012. URL https://www.sciencedirect.com/science/article/pii/S0022053105001808.

[22] Alexander Reffgen. Strategy-proof social choice on multiple and multi-dimensional single-peaked domains. *Journal of Economic Theory*, 157:349–383, 2015. ISSN 0022-0531. doi: https://doi.org/10.1016/j.jet.2014.12.010. URL https://www.sciencedirect.com/science/article/pii/S0022053114001872.

[23] Alvin E. Roth. Incentive compatibility in a market with indivisible goods. *Economics Letters*, 9(2): 127–132, 1982. ISSN 0165-1765. doi: https://doi.org/10.1016/0165-1765(82)90003-9. URL https://www.sciencedirect.com/science/article/pii/0165176582900039.

[24] Alvin E. Roth and Andrew Postlewaite. Weak versus strong domination in a market with indivisible goods. *Journal of Mathematical Economics*, 4(2):131–137, 1977. ISSN 0304-4068. doi: https://doi.org/10.1016/0304-4068(77)90004-0. URL https://www.sciencedirect.com/science/article/pii/0304406877900040.

[25] Alvin E Roth and Marilda A Oliveira Sotomayor. *Two-sided matching: A study in game-theoretic modeling and analysis*. Number 18. Cambridge University Press, 1992.

[26] Alvin E Roth, Tayfun Sonmez, and M Utku Unver. Kidney exchange. *Quarterly Journal of Economics*, 119(2):457–488, 2004.

[27] Jay Sethuraman. An alternative proof of a characterization of the ttc mechanism. *Operations Research Letters*, 44(1):107–108, 2016.

[28] Lloyd Shapley and Herbert Scarf. On cores and indivisibility. *Journal of mathematical economics*, 1 (1):23–37, 1974.

[29] Tayfun Sönmez. Strategy-proofness and essentially single-valued cores. *Econometrica*, 67(3): 677–689, 1999.

[30] Yuki Tamura. Object reallocation problems with single-dipped preferences. *Games and Economic Behavior*, 140:181–196, 2023. ISSN 0899-8256. doi: https://doi.org/10.1016/j.geb.2023.03.006. URL https://www.sciencedirect.com/science/article/pii/S0899825623000386.

[31] Yuki Tamura and Hadi Hosseini. The crawler: Three equivalence results for object (re)allocation problems when preferences are single-peaked. *Journal of Economic Theory*, 203:105466, 2022. ISSN 0022-0531. doi: https://doi.org/10.1016/j.jet.2022.105466. URL https://www.sciencedirect.com/science/article/pii/S0022053122000564.

[32] Peter Troyan. Obviously strategy-proof implementation of top trading cycles. *International Economic Review*, 60(3):1249–1261, 2019. doi: https://doi.org/10.1111/iere.12384. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/iere.12384.

8

List of Publication(s)/Submitted Article(s)

- **Published/Accepted Papers**

  (1) "Incentive-compatible voting rules with positively correlated beliefs": Correction *Theoretical Economics*, 2022, Volume 17, Number 2, 929-942.

  (2) "Ordinal Bayesian incentive-compatible voting rules with correlated belief under betweenness property" *Economic Letters*,2023, Volume 229, 111223.

- **Completed Papers**

  (1) "Social choice with network-based common knowledge".

  (2) "Matching under multiple single-peaked domains".

  (3) "Matching under tree single-peaked preferences".

  (4) "Matching rules and obvious strategy-proofness".