Housing Market on Networks

Xinwei Song
Key Laboratory of Intelligent Perception and Human-Machine Collaboration, ShanghaiTech University
Shanghai, China
songxw2024@shanghaitech.edu.cn

Tianyi Yang
Key Laboratory of Intelligent Perception and Human-Machine Collaboration, ShanghaiTech University
Shanghai, China
yty2@shanghaitech.edu.cn

Dengji Zhao
Key Laboratory of Intelligent Perception and Human-Machine Collaboration, ShanghaiTech University
Shanghai, China
zhaodj@shanghaitech.edu.cn

ABSTRACT

Incentivizing the existing participants to invite new participants to join an auction, matching or cooperative game have been extensively studied recently. One common challenge to design such incentive in these games is that the invitees and inviters are competitors. To have such an incentive, we normally have to sacrifice some of the traditional properties. Especially, in a housing market (one kind of one-sided matching), we cannot maintain the traditional stability and optimality. The previous studies proposed some new matching mechanisms to have the invitation incentive (part of the incentive compatibility), but did not have any guarantee on stability and optimality.

In this paper, we propose new notions of stability and optimality which are achievable with incentive compatibility. We weaken stability and optimality on a special structure (complete components) on networks. We first prove that the weakened notions are the best we can achieve with incentive compatibility. Then, we propose three mechanisms (Swap With Neighbors, Leave and Share, and Connected Trading Cycles) to satisfy the desirable properties. Connected Trading Cycles is the first mechanism to satisfy the best stability and optimality compatible with incentive compatibility.

KEYWORDS

Mechanism Design; Housing Market; Incentive Compatible

ACM Reference Format:
Xinwei Song, Tianyi Yang, and Dengji Zhao. 2025. Housing Market on Networks. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

1 INTRODUCTION

Housing market on networks studies an endowment exchange problem with special consideration of the participants' social connections [9]. In this model, participants' social connections are private information and only a small group initiates the matching. Those who are already in the game can decide whether or not to invite their neighbors. Expanding the market via invitation can be beneficial because more participants will provide more matching options, which opens the possibility of a more satisfactory matching. Several real-world applications, like online housing markets and second-hand goods exchange platforms, have utilized social networks to promote their markets. On these platforms, existing participants are encouraged to share the matching information on their social media and invite their friends to form larger markets. But sometimes participants may hesitate to invite because their friends will compete with them and take away their favorite items in the matching game. For this reason, the main obstacle is ensuring that the inviters' match is not getting worse after inviting others.

To solve the above problem, one way is to constrain participants' selection space to remove potential competition caused by invitation. For example, Kawasaki et al. [9] restricted the social network between agents as trees and proposed Modified TTC which only allows participants to choose from their parents and descendants. In this paper, we aim to design mechanisms for general social networks structures and design mechanisms to incentivize invitations. Our first mechanism, Swap With Neighbors (SWN) restricts participants to choosing from their neighbors' items. Under this restriction, invitations can only bring matching opportunities not competitions, thus naturally satisfying IC. The drawback of SWN is that there is a very low probability that a participant can benefit from the enlarged market, which contradicts the purpose of enlarging the market for better matching. To improve the matching, we observed that if a group of participants trade and leave the market, they do not care about what happens next in the matching. Inspired by this, our second mechanism, called Leave and Share (LS), uses SWN as a base protocol but dynamically connects the remaining participants when some are leaving the market (a sharing process). Due to the sharing process, LS gives a more satisfying outcome.

All the above-mentioned mechanisms focused on the IC property, but to properly evaluate a matching mechanism, optimality and stability are the two key properties. The former characterizes how efficient a matching is, i.e., whether participants can improve their matching without making others' matching worse, and the latter depicts the robustness of a matching, i.e., whether participants have incentives to deviate from the matching and exchange within a smaller group. The Top Trading Cycles (TTC) presented by Shapley and Scarf [14] is the only Pareto optimal and stable mechanism for housing market problem [13]. However, Kumar et al. [10] show that more than half of the participants prefer their matching given by TTC in a smaller market than in a large integrated market. Hence, in the network setting where participants can control the market size through strategic invitations, TTC fails to be incentive compatible (IC). Due to the uniqueness of TTC, stability and Pareto optimality are not compatible with IC in housing market on networks.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

Table 1: Comparison on housing market mechanisms on social networks. We propose SWN, LS and CTC in this paper.

Mechanism	*Stable-cc*	*Optimal-cc*	*IC*	*IR*
TTC [14]	✓	✓	×	✓
Modified TTC [9]	✓(Trees)	×	✓(Trees)	✓
SWN	✓	×	✓	✓
LS	✓	×	✓	✓
CTC	✓	✓	✓	✓

To maintain invitation incentives, we weaken stability and optimality notions with complete components (cc), called stable-cc and optimal-cc, respectively. Stable-cc weakens stability in the sense that only participants in a complete component can deviate from the matching and exchange within the component. Optimal-cc weakens optimality in the sense that a matching can only be improved if all the participants whose allocation is improved have to form a complete component. We then prove that both stable-cc and optimal-cc are the tightest notions compatible with IC. Besides IC, individual rationality (IR) is also a commonly considered property that ensures participating in the game is not harmful, serving as a base incentive for people to join the game. Finally, combining all the properties together, we construct the theoretical boundaries for housing market on networks.

The best properties we can achieve together for our setting are stable-cc, optimal-cc, IC, and IR. The previously mentioned SWN and LS detect all the cycles formed by neighbors, so they naturally satisfy stable-cc. However, they both add restrictions to the selection space and fail to satisfy optimal-cc. Hence, we further propose the third mechanism, called Connected Trading Cycles (CTC), which does not add restrictions on the participant's selection space to satisfy all the properties. Similar to TTC, CTC detects the top trading cycles (to ensure optimality). The key difference is that CTC makes sure the trading cycles have to satisfy certain connectedness on networks (to ensure invitation incentives). Intuitively, the connectedness guarantees that a group of participants can stay together regardless of the others' strategic invitations. Thus, in the network setting, the trading cycles should build on a group of connected participants.

To sum up, our contributions are threefold:

- We define the stability and optimality notions compatible with IC for housing market on networks.
- We prove the best achievable stability and optimality in the network setting.
- We propose three mechanisms: Swap With Neighbors, Leave and Share, and Connected Trading Cycles to meet the desirable properties (Table 1 gives a quick overview of the theoretical performance of each mechanism).

2 RELATED WORK

Li et al. [12] initiate the line of research on mechanism design on social networks. They utilize social networks to incentivize participants to invite their friends to form a larger market so that participants can receive better outcomes [17]. However, the participants are competitors and may not want to invite all their friends without proper incentives. Many classic solutions fail to provide such incentives, so we cannot directly apply them for this purpose. To combat this, in auctions, we can pay the harmed inviters some rewards [11]. In cooperative games, we can let the inviters share their invitees' contributions [16]. However, in matching, we face a greater challenge since the mechanism cannot compensate participants for the loss caused by invitation through payments.

Existing solutions for matching on networks restricts participants' selection space and the social network structures. Kawasaki et al. [9] and You et al. [15] model the social network as a directed graph where neighborhood relationships can be asymmetric (i.e., A is B's neighbor does not necessarily mean B is A's neighbor). They focus on trees and modify the classic mechanisms to incentivize invitations for housing market as well as its variant [1]. Specifically, they add limits on the selection space of the participants in a tree to prevent an invitee competing with her inviters. For two-sided matching, Cho et al. [5] model the school choice problem on networks and design invitation incentives for the student side only. Another thread of work also takes into account the social network and its influence, but in their setting, the social network is priorly known and it constrains possible allocations [7, 8].

Optimality and stability are two an well-concerned property for matching mechanisms [2]. The celebrated TTC is the only Pareto optimal and stable solution in the traditional housing market problem [13]. Abraham et al. [3] study various ways to evaluate the optimality of a house allocation mechanism and illustrate why TTC gives the Pareto optimal allocation. Fleischer and Wang [6] point out the transitions between different optimal allocations in the house allocation problem. Brandt and Wilczynski [4] analyze whether different types of dynamic pairwise swaps converge to Pareto optimal allocations for matching markets. In the network setting, no previous work has investigated the theoretical boundaries. We propose tight stability and optimality notions for this new setting and design mechanisms to reach them.

3 THE MODEL

We consider a housing market problem on a social network denoted by an undirected graph $G = (N, E)$, which contains n agents $N = \{1, \ldots, n\}$. Each agent $i \in N$ is endowed with an indivisible item h_i, usually referred to as a house. Let $H = \{h_1, \ldots, h_n\}$ be the set of all agents' items. Considering the nature of the exchange economy, we formulate agents' social relationships as symmetric in this model, i.e., i has connections with j in the social network indicates j has connections with i. We define agent i as j's neighbor if there is an edge $e \in E$ between agent i and j, let $r_i \subseteq N$ be i's neighbor set.

Despite social relationship, each agent $i \in N$ has a strict preference $\succ_i$ over H. $h \succ_i h'$ means i prefers h to h' and we use $\succeq_i$ to represent the weak preference. Thus, we denote agent i's private type as $\theta_i = (\succ_i, r_i)$ and $\theta = (\theta_1, \cdots, \theta_n)$ as the type profile of all agents. Let θ_{-i} be the type profile of all agents except for agent i, then θ can be written as (θ_i, θ_{-i}). Let Θ be the type profile space of all agents. Similarly, we have $\Theta = (\Theta_i, \Theta_{-i})$.

In the housing market problem, the goal is to construct a matching following the principle that each agent will exchange their endowments to get better allocations. Assume there is a trusted center to enforce the execution of a matching mechanism, each agent is required to report her type (reporting neighbor set is treated as inviting neighbors in practice). We denote agent i's reported type as $\theta'_i = (\succ'_i, r'_i)$, where $\succ'_i$ is the reported preference and $r'_i \subseteq r_i$ is the reported neighbor set. Let $\theta' = (\theta'_1, \cdots, \theta'_n)$ be the reported type profile of all agents.

Definition 3.1. In a housing market problem, a matching mechanism is defined by an allocation policy $\pi = (\pi_i)_{i \in N}$, where $\pi_i : \Theta \to H$ satisfies for all $\theta \in \Theta$, $\pi_i(\theta) \in H$, and $\pi_i(\theta) \neq \pi_{j \neq i}(\theta)$.

In the networked setting, we assume only a subset of the agents are initially in the game and the matching mechanism can incentivize agents to diffuse the information thus enlarging the matching. Without loss of generality, suppose an agent set $N_0 \subseteq N$ contains the initial participants in the market. The others need the existing participants' invitation to join the game. Since the invitation is modeled as reporting neighbors, we define the qualified participants by their reported types.

For a given reported type profile θ', we generate a directed graph $G(\theta') = (N(\theta'), E(\theta'))$, where edge $\langle i, j \rangle \in E(\theta')$ if and only if $j \in r'_i$. Under θ', we say agent i is qualified if and only if there is a path from any agent in N_0 to i in $G(\theta')$. That is, i can connect to the agent set N_0 by an invitation chain. Let $Q(\theta')$ be the set of all qualified agents under θ'. In the networked housing market problem, diffusion matching mechanisms can only use $Q(\theta')$.

Definition 3.2. In a networked housing market problem, a diffusion matching mechanism is a matching mechanism $\pi = (\pi_i)_{i \in N}$, such that for all reported type profile θ', it satisfies:

(1) for all unqualified agents $i \notin Q(\theta')$, $\pi_i(\theta') = h_i$.
(2) for all qualified agents $i \in Q(\theta')$, $\pi_i(\theta')$ is independent of the reports of all unqualified agents.

Different from Definition 3.1, in a diffusion matching, the participants can affect the qualification of other participants. If a participant changes her reported neighbor set, the qualified agent set may change. This is the challenge of this setting.

Next, we define two desirable properties for diffusion matching mechanisms: individual rationality and incentive compatibility. Intuitively, individual rationality requires that for each agent, reporting her type truthfully guarantees that she gets an item no worse than her own. For incentive compatibility, it means reporting type truthfully is a dominant strategy for each agent.

Definition 3.3 (Individual Rationality (IR)). A mechanism π is individually rational if for all $i \in N$, all $\theta_i \in \Theta_i$, and all $\theta'_{-i} \in \Theta_{-i}$, we have $\pi_i(\theta_i, \theta'_{-i}) \succeq_i h_i$.

Definition 3.4 (Incentive Compatibility (IC)). A mechanism π is incentive compatible if for all $i \in N$, all $\theta'_{-i} \in \Theta_{-i}$ and all $\theta_i, \theta'_i \in \Theta_i$, we have $\pi_i(\theta_i, \theta'_{-i}) \succeq_i \pi_i(\theta'_i, \theta'_{-i})$.

To evaluate the performance of a matching mechanism, an important metric is Pareto optimality. It requires that no agent can improve her matching without others' allocation getting worse.

Definition 3.5 (Pareto Optimality (PO)). A mechanism π is Pareto optimal if for all type profile θ, there is no other allocation $\pi'(\theta)$ such that for each qualified agent $i \in Q(\theta)$, $\pi'_i(\theta) \succeq_i \pi_i(\theta)$, and there exists at least one qualified agent $j \in Q(\theta)$, $\pi'_j(\theta) \succ_j \pi_j(\theta)$.

Table 2: Desirable properties for diffusion matching mechanisms. IC, IR, stable-cc and optimal-cc are the boundaries.The implications between properties are proved in Section 4.3.

IR	*IC*	*Stability*	*Optimality*
IR	IC	Stable	PO
		↓	↓
		Stable-wcc	Optimal-wcc
		↓	↓
		Stable-cc	Optimal-cc

Another metric is stability. It requires no group of agents can deviate from the matching and match among themselves to make no one in the group worse off and at least one better off.

Definition 3.6 (Stability). A mechanism π is stable if for all type profile θ, there is no other allocation $\pi'(\theta)$ such that there exists a group $S \subseteq Q(\theta)$ that satisfies $\forall i \in S, \pi'_i(\theta) \in H_S$, and $\forall i \in S, \pi'_i \succeq_i \pi_i(\theta)$ with at least one $j \in S, \pi'_j(\theta) \succ_j \pi_j(\theta)$.

In summary, our model is a practical variant of the networked housing market model proposed in [9]. In their paper, the authors designed matching on an asymmetric social network and proved the impossibility results of the compatibility of IC, PO and stability. In the next section, we will prove the impossibilities in our model and further construct the theoretical boundaries.

4 THE BOUNDARIES

In this section, we first prove that a diffusion matching mechanism cannot achieve Pareto optimality or stability together with IC. Next, we define the best optimality and stability notion compatible with IC and then construct the theoretical boundaries.

4.1 Impossibility Results

THEOREM 4.1 (IMPOSSIBILITY FOR PO, IC AND IR). *No diffusion matching mechanism is PO, IC and IR.*

PROOF. In the example shown in Figure 1, suppose $N_0 = \{1, 2\}$, the Pareto optimal and IR allocations are $\pi_1 = (h_3, h_2, h_1)$ and $\pi_2 = (h_2, h_1, h_3)$. If a mechanism allocates π_1, agent 2 can misreport her neighbor set as $r'_2 = \{1\}$. Under agent 2's misreport, agent 3 cannot join the game and the only PO and IR allocation will be $\pi_2 = (h_2, h_1, h_3)$, and 2 gets a better allocation compared to that in π_1. However, if the mechanism allocates π_2, agent 1 can misreport her preference as $h_3 \succ'_1 h_1 \succ'_1 h_2$. In this way, the only PO and IR allocation is $\pi_1 = (h_3, h_2, h_1)$, and agent 1 reaches a better allocation. Hence, no diffusion matching mechanism is PO, IC and IR. □

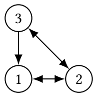


Figure 1: A social network example. Preferences are $h_3 \succ_1 h_2 \succ_1 h_1$, $h_1 \succ_2 h_2 \succ_2 h_3$, $h_1 \succ_3 h_3 \succ_3 h_2$.

Theorem 4.2 (Impossibility for stability and IC). *No diffusion matching mechanism is stable and IC.*

Proof. Consider the example in Figure 1, suppose $N_0 = \{1\}$, the only stable allocations is $\pi_1 = (h_3, h_2, h_1)$. However, agent 2 can misreport her neighbor set as $r'_2 = \{1\}$, so agent 3 cannot join the game. In this way, the only stable allocation is $\pi_2 = (h_2, h_1, h_3)$. This means agent 2 can misreport to improve her matching result, so stability is incompatible with IC. □

To seek achievable optimality and stability notions for IC diffusion matching mechanisms, we focus on a special graph structure: complete components, i.e., the agents are neighbors of each other. For Pareto optimality, we restrict that only fully connected agents can improve their matching. For stability, we constrain the agents who can deviate from the matching and swap among themselves to be fully connected.

Definition 4.3 (Complete Component). A connected directed graph $G = (V, E)$ is a complete component if for any two nodes $i, j \in V$, we have $\langle i, j \rangle \in E$.

The idea of the limitation comes from the reality that only a group of people who know each other have a higher possibility to communicate and negotiate for a better matching/trade. Besides, in the traditional housing market problem, since there are no constraints on social connections, agents can be viewed as fully connected. Then, any group of agents is a fully connected component.

Following this idea, we define optimality and stability under complete components: optimal-cc and stable-cc.

Definition 4.4 (Optimality under Complete Components (Optimal-cc)). A diffusion matching mechanism π is optimal under complete components if for all type profiles θ and the allocation $\pi(\theta)$, there is no other allocation $\pi'(\theta)$ such that $\forall i \in N, \pi'_i(\theta) \succeq_i \pi_i(\theta)$ and $\exists j \in N, \pi'_j(\theta) \succ_j \pi_j(\theta)$ and agents $\{i \in N | \pi_i(\theta) \neq \pi'_i(\theta)\}$ forms a **complete component** in $G(\theta)$.

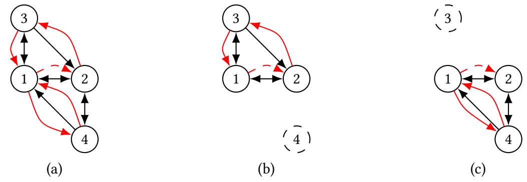
Definition 4.5 (Stability under Complete Components (Stable-cc)). A diffusion matching mechanism π is stable under complete components if for all type profiles θ and the allocation $\pi(\theta)$, there is no agent set $S \subseteq N$ (with item set $H_S \subseteq H$) that forms a **complete component** in $G(\theta)$, and another allocation $\pi'(\theta)$ with $\forall i \in S, \pi'_i(\theta) \in H_S$ such that $\forall i \in S, \pi'_i(\theta) \succeq_i \pi_i(\theta)$ and $\exists j \in S, \pi'_j(\theta) \succ_j \pi_j(\theta)$.

4.2 Stable-cc and Optimal-cc are the Boundaries

Stable-cc and optimal-cc pose strict constraints on the agents who can improve by exchanging internally. Hence, it is worth investigating whether we can relax this limitation to get stronger notions. For instance, can we allow any connected group (no need to be a complete component) to improve their matching? Following this idea, we define stronger notions based on weakly complete components, but they are not compatible with IC.

Definition 4.6 (Weakly Complete Component). A connected directed graph $G = (V, E)$ is a weakly complete component if there exists at most one pair of nodes $i, j \in V$ such that $\langle i, j \rangle \notin E$.

Definition 4.7 (Optimality under Weakly Complete Components (Optimal-wcc)). A diffusion matching mechanism π is optimal under weakly complete components if for all type profiles θ and the allocation $\pi(\theta)$, there is no other $\pi'(\theta)$ such that $\forall i \in N, \pi'_i(\theta) \succeq_i \pi_i(\theta)$ and $\exists j \in N, \pi'_j(\theta) \succ_j \pi_j(\theta)$ and agents $\{i \in N | \pi_i(\theta) \neq \pi'_i(\theta)\}$ forms a **weakly complete component** in $G(\theta)$.



Example	*Optimal-wcc Allocation*
(a)	$\pi_a = (h_4, h_2, h_3, h_1), \pi_b = (h_2, h_3, h_1, h_4)$
(b)	$\pi_c = (h_2, h_3, h_1, h_4)$
(c)	$\pi_d = (h_4, h_2, h_3, h_1)$

Figure 2: A counterexample for the coexistence of optimal-wcc, IR, and IC. Preferences are $h_4 \succ_1 h_2 \succ_1 h_1$, $h_3 \succ_2 h_2$, $h_1 \succ_3 h_3$, $h_1 \succ_4 h_4$. Agents 1,2 are initial players in the matching. The red solid arrows represent favorite pointing. The red dashed arrows represent the second favorite pointing. The dashed agents are unqualified, so they are allocated their endowments.

Next, we prove that no IC and IR diffusion mechanisms can be optimal-wcc.

Theorem 4.8. *No IC and IR diffusion one-sided matching mechanism is optimal-wcc.*

Proof. Let's consider the example given in Figure 2. In figure (a), IR and optimal-wcc solutions are π_a, π_b. If we choose π_a, agent 2 cannot get h_3. So, agent 2 has the incentive to misreport and not invite agent 4. In this way, figure (a) transforms to figure (b) where the only IR and optimal-wcc solution is π_c. Then, agent 2 can get her favorite item h_3. To ensure IC, we should allocate h_3 to agent 2 when she truthfully reports (i.e., allocate π_b in figure (a)). Next, we consider agent 1's incentive to misreport when we choose π_b in figure (a). If agent 1 does not invite agent 3, figure (a) transforms to figure (c) where the only IR and optimal-wcc solution is π_d. Here, agent 1 can get her favorite item h_4. To ensure IC in figure (a), we should allocate h_4 to agent 1 when she truthfully reports (i.e., allocate π_a in figure (a)). This is a contradiction, so optimal-wcc is not compatible with IC. Hence, the tightest optimality notion compatible with IC is optimal-cc. □

Similarly, for stability, we define a stronger notion called stable-wcc and prove that stable-wcc is not compatible with IC and IR.

Definition 4.9 (Stability under Complete Components (Stable-wcc)). A diffusion matching mechanism π is stable under weakly complete components if for all type profiles θ and the allocation $\pi(\theta)$, there is no agent set $S \subseteq N$ (with item set $H_S \subseteq H$) that forms a **weakly complete component** in $G(\theta)$, and another allocation $\pi'(\theta)$ with $\forall i \in S, \pi'_i(\theta) \in H_S$ such that $\forall i \in S, \pi'_i(\theta) \succeq_i \pi_i(\theta)$ and $\exists j \in S, \pi'_j(\theta) \succ_j \pi_j(\theta)$.

Theorem 4.10. *No IC and IR diffusion one-sided matching mechanism is stable-wcc.*

Proof. Consider the example in Figure 1, suppose $N_0 = \{1\}$, the only stable-wcc allocations is $\pi_1 = (h_3, h_2, h_1)$. However, agent 2 can misreport her neighbor set as $r_2' = \{1\}$, so that agent 3 cannot join the game. In this way, the only stable-wcc allocation is $\pi_2 = (h_2, h_1, h_3)$. This means agent 2 can misreport to improve her matching result, thus, stable-wcc is incompatible with IC. □

Combining Theorem 4.8 and Theorem 4.10, we conclude that the best stability and optimality we can aim for an IC and IR diffusion matching mechanism are stable-cc and optimal-cc respectively.

4.3 Implications between Properties

In this section, we study the relationships between different stability and optimality notions. By definition, Pareto optimality implies optimal-wcc and further implies optimal-cc. Stability implies stable-wcc and further implies stable-cc. In the traditional setting, a stable matching mechanism naturally satisfies Pareto optimal.

Theorem 4.11. *A mechanism π is stable implies that π is PO.*

Proof. For all type profiles θ, if a stable mechanism π is not Pareto optimal, there exists a π' such that $\forall i \in N, \pi_i'(\theta) \succeq_i \pi_i(\theta)$ and $\exists j \in N, \pi_j'(\theta) \succ_j \pi_j(\theta)$. In this case, we can construct a group of agents S starting from $\{j \in N | \pi_j'(\theta) \succ_j \pi_j(\theta)\}$ and consecutively add other agents so that $\forall i \in S, \pi_i'(\theta) \in H_S$. So, S can deviate from the matching together which violates stability. Thus, π is stable implies that π is Pareto optimal. □

Thus, when designing mechanisms, we can only focus on stability, and the optimality condition will be satisfied at the same time. However, in the networked setting, a similar implication relationship does not exist between the redefined notions.

Theorem 4.12. *A mechanism π is stable-cc does not imply that π is optimal-cc.*

Proof. See the example in Figure 3, allocation $\pi = (h_2, h_1, h_4, h_3)$ is stable under complete components. However, there exists another allocation $\pi' = (h_2, h_4, h_1, h_3)$ where $\{2, 3\}$ is a strictly better off group. Thus, π is stable-cc does not imply that π is optimal-cc. □

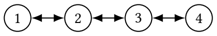


Figure 3: Preferences are $h_3 \succ_1 h_2 \succ_1 h_1$, $h_4 \succ_2 h_1 \succ_2 h_2$, $h_1 \succ_3 h_4 \succ_3 h_3$, $h_2 \succ_4 h_3 \succ_4 h_4$.

This vanish of implication also poses a greater challenge in designing a mechanism that reaches the new boundaries.

5 THE MECHANISMS

In this section, we introduce three IC mechanisms, Swap With Neighbors (SWN), Leave and Share (LS) and Connected Trading Cycles (CTC). By different techniques, they all guarantee that inviting more agents to join in only potentially improves one's matching[1].

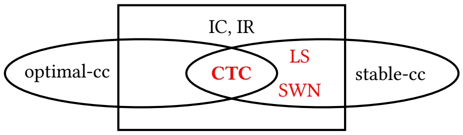


Figure 4: The properties of our three mechanisms. LS and SWN are stable-cc, IC and IR. CTC reaches the boundaries.

5.1 Swap With Neighbors

First, we present SWN which is an intuitive extension of Top Trading Cycles. SWN only allows agents to choose favorite items from neighbors to achieve invitation incentives.

Definition 5.1 (Swap With Neighbors). For a given $G(\theta')$, construct a directed graph by letting each agent point to her favorite item among herself and her neighbors remaining in the matching. There is at least one cycle. For each cycle, allocate the item to the agent who points to it and remove the cycle. Repeat the process until there is no agent left.

In SWN, agents' allocation is determined by trading cycles, which makes it strategy-proof for the preference report. Also, each agent can only get allocated a house from her neighbors, which means misreporting on one's neighbor set is not beneficial, so SWN satisfies IC. Besides, a trading cycle within neighbors can always be allocated by SWN, so SWN satisfies stable-cc.

However, SWN achieves IC and stable-cc at the cost of efficiency, as it limits the matching options for agents. To improve the matching, we should provide more matching opportunities. But this also brings agents' strategic report aiming for a better match. Thus, we need a strategy-proof order to help with the mechanism design and ensure that truthfully reporting one's type is a dominant strategy for every agent. Our order is based on the shortest path length from the initial agent set to each agent. The reason is that non-initial agents require other agents' permission to join the matching, the layer structure of the social network should be kept properly for incentive compatibility. This order is applied in both LS and CTC.

Definition 5.2. An ordering of agents is a one-to-one function $O : \mathbb{N}^+ \to N$, where agent $O(t)$ is the t^{th} agent in the ordering. For simplicity, we denote agent i's order as $O^{-1}(i)$. Agents in O are sorted in ascending order by the length of the shortest path from agent set N_0 to them. Especially, for any agent $i \in N_0$, its shortest path length from agent set N_0 is 0. When multiple agents have the same length of the shortest path, we use a random tie-breaking.

5.2 Leave and Share

Leave and Share uses SWN as a base and adds a natural sharing process to enlarge agents' selection space, trying to provide a better allocation. The mechanism consists of two steps: first, agents are matched and left by rounds in a protocol that resembles SWN under a strategy-proof order. This guarantees that inviters are not worse off. Second, we share the neighbors of the left agents in this round by connecting their neighbors to each other, thus their neighbors can have new neighbors in the next round. This dynamic neighbor set update comes naturally because a matched cycle does not care

[1] Refer to our arXiv version for the omitted proofs and simulation results.

how the remaining neighbors will be matched. Also, their remaining neighbors and other agents cannot prevent this sharing.

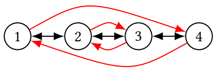


Figure 5: Preferences are $h_4 \succ_1 h_1 \succ_1 \cdots$, $h_3 \succ_2 h_2 \succ_2 \cdots$, $h_2 \succ_3 h_3 \succ_3 \cdots$, $h_1 \succ_4 h_4 \succ_4 \cdots$.

To see the value of our mechanism, consider the example given in Figure 5, where only agents 2 and 3 can exchange with each other in SWN. The rest of the agents will end up with their own items. However, agents 2 and 3 will not block the exchange for agents 1 and 4 once they get their preferred items. After agents 2 and 3 are matched and **Leave**, we **Share** their remaining neighbors then agents 1 and 4 can swap. The process of Leave and Share is the name and core of our mechanism. Before formalizing our mechanism, we introduce two notations to simplify our description.

Definition 5.3. Given a set $A \subseteq N$, we say $f_i(A) = j \in A$ is i's favorite agent in A if for any agent $k \in A, h_j \succeq_i' h_k$.

```
Leave and Share (LS)
  (1) Initialize $N_{out} = \emptyset$ and an empty stack $S$. Define the top and bottom of $S$ as $S_{top}$ and $S_{bottom}$ respectively, and let $R_i = r_i' \cup \{S_{bottom}, i\}$.
  (2) While $N_{out} \neq N$:
    (a) Find the minimum $t$ such that $O(t) \notin N_{out}$. Push $O(t)$ into $S$.
    (b) While $S$ is not empty:
      (i) While $f_{S_{top}}(R_{S_{top}}) \notin S$, push $f_{S_{top}}(R_{S_{top}})$ into $S$.
      (ii) Pop off all agents from $S_{top}$ to $f_{S_{top}}(R_{S_{top}})$, who already formed a trading cycle $C$ following their favorite agents. Allocate each agent $i \in C$ the item $h_{f_i(R_i)}$ . Add $C$ to $N_{out}^t$.
      (iii) Update the neighbor set of $C$'s remaining neighbors by removing $C$, i.e., for all $j \in \bigcup_{i \in C} r_i' \setminus N_{out}^t$, set $r_j' = r_j' \setminus C$.
    (c) Add $N_{out}^t$ to $N_{out}$. Let all remaining neighbors of $N_{out}^t$ connect with each other, i.e., they become neighbors of each other. That is, let $X = \bigcup_{i \in N_{out}^t} r_i' \setminus N_{out}^t$ and for all $j \in X$, set $r_j' = r_j' \cup X$.
```

In LS, we first define an order O which depends on each agent's shortest distance to the initial agent set. Under this order, the first while loop (step 2) guarantees that the agent pushed into the stack is the remaining agent with the smallest order, and all agents are matched (including self-match) in the end. A new round begins each time the stack empties.

In the Leave stage, each agent that is pushed into the stack pushes her (current) favorite neighbor into the stack (step (a)). If her favorite is already in the stack, we pop all the agents between herself and her favorite to form a trading cycle. Specially, we allow agents to choose the agent at the bottom of the stack as favorite, which leads to popping off all the agents in the stack (step (b)).

Once the stack is empty, the mechanism enters the Share stage and updates the neighbor set of the remaining agents (step (c)). All neighbors of the left agents become new neighbors to each other. In the next Leave stage, they can choose in a larger neighbor set.

5.3 Connected Trading Cycles

In this section, we present our mechanism called Connected Trading Cycles (CTC) which takes both the trading cycles and agents' connections into consideration. CTC is IR, IC, optimal-cc, and stable-cc, which is the best we can get in the network setting. To begin with, we give a few definitions to serve the description of our mechanism.

Definition 5.4. Given a reported type profile θ', we generate a directed graph $F(\theta')$, in which each qualified agent has exactly one outgoing edge pointing to a qualified agent in $G(\theta')$. An edge $\langle i, j \rangle$ in $F(\theta')$ means agent i likes agent j's item the most among all the qualified agents' items (i, j can be the same agent). We name $F(\theta')$ as the favorite pointing graph for θ'.

There is at least one cycle in $F(\theta')$, and we formalize the definition of a cycle as below.

Definition 5.5. A cycle in $F(\theta')$ is an agent sequence $C_m = \{c_1, \ldots, c_m\}$ such that $\forall i \in \{1, \ldots, m-1\}$, c_i points to c_{i+1}, and c_m points to c_1 in the favorite pointing graph $F(\theta')$.

Since the optimality and stability boundaries are constructed on complete components, it is natural to let cycles build on complete components to trade. However, if we only allow cycles formed by complete components to get traded, the performance is not better than SWN or LS (it meets stable-cc, but far from optimal-cc). Thus, we also consider cycles formed by connected components. Given that optimal-c is not compatible with IC, it is clear that not all such cycles can get traded. Therefore, we should further distinguish these cycles and identify the ones compatible with IC. This detection also makes our CTC description complex.

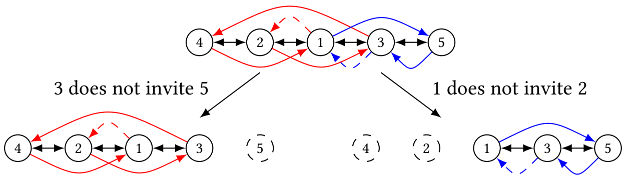


Figure 6: Preferences are $h_5 \succ_1 h_2 \succ_1 h_1$, $h_3 \succ_2 h_2$, $h_4 \succ_3 h_1 \succ_3 h_3$, $h_1 \succ_4 h_4$, $h_3 \succ_5 h_5$. The solid arrows represent favorite pointing. The dashed arrows represent second favorite pointing. The dashed agents are unqualified agents.

Here comes a motivated example. The social network is shown in Figure 6 and agent 1 is the initial agent. If agent 1 does not invite agent 2, the connected component $\{1, 3, 5\}$ will form the blue cycle ($C_3 = \{1, 5, 3\}$). On the other hand, if agent 3 does not invite agent 5, the connected component $\{4, 2, 1, 3\}$ will form the red cycle ($C_4 = \{1, 2, 3, 4\}$). If both cycles are allowed, IC cannot hold since both agent 1 and agent 3 have the incentive to not invite a neighbor and form the cycle in favor of herself. Thus, an IC mechanism can only allow at most one of these two cycles to get traded. To distinguish the two cycles, we pay attention to each agent's pointing. In the blue cycle, each agent has an exclusive path to reach her pointing: **1** ⟶ 3, **3** ⟶ 5, **5** ⟶ 3 ⟶ 1. However, in the red cycle, **2** ⟶ 1 ⟶ 3 and **4** ⟶ 2 ⟶ 1 have a shared edge 2 ⟶ 1. Similarly, **1** ⟶ 2 and **3** ⟶ 1 ⟶ 2 ⟶ 4 have a shared edge 1 ⟶ 2. Recall that in the cycle formed by a complete component (which is always allowed to get traded), each agent has an exclusive path to her pointing. Intuitively, the exclusive path can be regarded as an insurance for invitation incentives (no agent wants to influence other's pointing by not inviting a neighbor). Following this idea, we design an algorithm to detect cycles in which each agent has an exclusive path to her pointing.

We also needs a strategy-proof order of the agents to determine who can be matched first.

Path Detection

Input: a cycle C_m, a reported type profile θ', a favorite pointing graph $F(\theta')$ and social network $G(\theta')$.

a. Find a minimum connected component T in $G(\theta')$ where $C_m \subseteq T$ and $\forall i \in T$, let j be i's pointing in $F(\theta')$, we have $j \in T$. Note that if the cycle C_m form a connected component in $G(\theta')$, then $T = C_m$. If there is no such T let $T = \emptyset$.

b. Construct a subgraph G_T with the reported type profile of all agents in T.

c. If agent $i \in T$ points to herself in $F(\theta')$, add a mark i on all the outgoing edges of i in G_T.

d. Construct a shortest path set $SP = \{sp_i | i \in T\}$ in G_T, where sp_i is the shortest path from agent $i \in T$ to her pointing in $F(\theta')$. While $SP \neq \emptyset$, execute the following,

   i. Sort SP by the ascending order of path length. If sp_i and sp_j have the same length, sort them by the agents ascending order $O^{-1}(i)$ and $O^{-1}(j)$.

   ii. Remove the first sp_i in SP. Add a mark i on every edge on path sp_i in G_T. If sp_i contains marks other than i, and i has another path to her pointing, add the next shortest path sp'_i for i to SP.

Output: an agent set T and the marked subgraph G_T.

We use a brief example to show how path detection works. As shown in Figure 7(a), agents' social connections (edges in $G(\theta')$) are denoted by double arrows, and the red arrows represent edges in the favorite pointing graph $F(\theta')$. Take cycle $C_5 = \{1, 5\}$ as an example, the minimum connected component for C_5 is $T = \{1, 2, 4, 5\}$. Agent 4 is involved to help agent 1 and 5 connect to each other, and since $\langle 4, 2\rangle \in F(\theta')$, agent 2 should also be involved in T (as in Path Detection step a). Then we detect the shortest path set SP for subgraph G_T, the sorted path set is shown in Figure 7(c). Finally, follow the instruction in Path Detection step d.ii, we mark the subgraph G_T to represents which directed edge by which agent.

In cycles like $C_4 = \{1, 2, 3, 4\}$ or $C_5 = \{1, 5\}$, not all agents have an exclusive path to their pointing, so some have to switch their pointing in $F(\theta')$ to a less preferred one. We define a next favorite function to adjust $F(\theta')$ based on agents' preferences.

Definition 5.6. Given a reported type profile θ', the qualified agent set is $Q(\theta')$. Suppose agent i points to agent j in the favorite pointing graph $F(\theta')$, we have $\succ_i^{next} (F(\theta')) = j'$ if j' owns i's favorite item in $Q(\theta') \setminus \{k \in Q(\theta') | k \succeq_i j\}$.

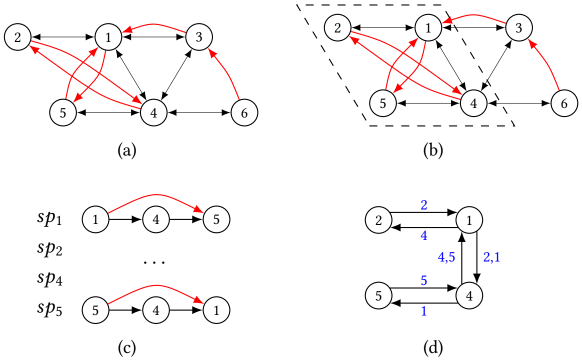


Figure 7: Path Detection process for cycle $C_5 = \{1, 5\}$. The red arrow is agents' pointing in the favorite pointing graph. The dashed box shows the minimum connected component T.

Now we are ready to introduce our mechanism.

Connected Trading Cycles (CTC)

Given a reported type profile θ', construct the reported social network $G(\theta')$ and favorite pointing graph $F(\theta')$. Initiate a settled agent set $V = \emptyset$. Execute the following until $V = N$.

a. Find the agent $p_1 \in N \setminus V$ with the minimum order $O^{-1}(p_1)$. Start from agent p_1, detect a node sequence $P_m = \{p_1, \dots, p_l, \dots, p_m\}$ in $F(\theta')$ such that $\forall i \in \{1, \dots, m-1\}$, p_i points to p_{i+1}, and $\{p_l, \dots, p_m\}$ is a cycle C_m.

b. Run **Path Detection** algorithm with input C_m, θ', $G(\theta')$ and $F(\theta')$. Get an agent set T and marked subgraph G_T as output.

c. Find every agent $i \in T$ such that no path in G_T from i to her pointing is exclusively marked by i, and i's outgoing edges are marked only by i. Denote the set of such agents as $S \subset T$.

   i. If $T = \emptyset$, find the last agent i on C_m who cannot connect to her pointing in $G(\theta')$, let i switch her pointing to $\succ_i^{next} (F(\theta'))$.

   ii. If $S = \emptyset$ and $C_m = T$, add all agents in T into the settled agent set V. While an agent $i \in N \setminus V$ points to an agent in V, let i point to $\succ_i^{next} (F(\theta'))$.

   iii. If $S = \emptyset$ and $C_m \subset T$, find the agent $j \in T \setminus C_m$ with minimum order $O^{-1}(j)$. Start from agent j's pointing, find the last agent $i \in C_m$ such that i's path to its pointing passes at least one agent in $T \setminus C_m$. If no such i, start from agent j's pointing, find the last agent $i \in (T \setminus C_m)$ such that i's path to its pointing passes at least one agent in C_m. Let i point to $\succ_i^{next} (F(\theta'))$.

   iv. If $(T \backslash C_m) \cap S \neq \emptyset$, find the agent $i \in (T \backslash C_m) \cap S$ with minimum order $O^{-1}(i)$. Let i point to $\succ_i^{next} (F(\theta'))$.

v. If $(T \setminus C_m) \cap S = \emptyset$, find the agent $i \in S$ with the minimum order $O^{-1}(i)$ such that i's path to its pointing covers another agent's path to her pointing. If no such i, find agent $i \in S$ with the minimum order $O^{-1}(i)$. Let i point to $\succ_i^{next}$ $(F(\theta'))$.

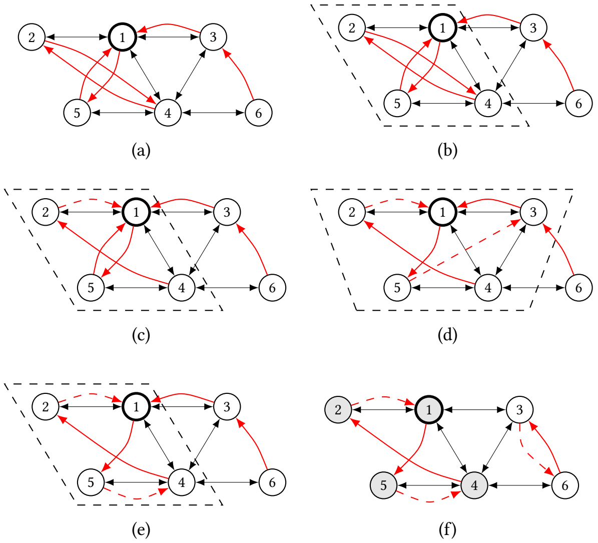
Our mechanism first detects cycles in the favorite pointing graph because these cycles can ensure optimality. Thus, agents will not misreport their preferences to get a better allocation (also why TTC is IC in the traditional setting). However, in the network setting, trading cycles formed by agents who cannot connect to each other are fragile. The reason is that others can misreport neighbor set to break the cycle (for instance, disqualify some agents on the cycle). Hence, we construct a minimum connected component T for each cycle where they can stay connected.

Recall the example in Figure 6, we can see that if every agent on the cycle has an exclusive path to her pointing (e.g. 1-3-5), the cycle can get traded. For cycles that do not meet this condition, we identify a special agent set S. each agent in S does not have an exclusive path to her pointing, and her outgoing edges are only marked by herself. These agents cannot misreport to avoid path overlap because the overlap lies outside their outgoing edges (which is irrelevant to their report). Hence, we let agents in S switch their pointing to a less preferred one.

For cases where $S = \emptyset$, if T consists of only agents on the cycle, they can get traded (CTC step c-ii). Otherwise, we start from an agent $i \in T$ who is not on the cycle and find the last agent on the cycle who does not have a path to her pointing only by connections on the cycle (CTC step c-iii). This is because every agent in T now has an exclusive path to her pointing, and the cycle has to rely on agents not on the cycle to stay connected. Thus, agents on the cycle have to switch preferences first, and we start from the last agent to maintain the completeness of the cycle to the greatest extent.

For cases where $S \neq \emptyset$, if S contains an agent not on the cycle, let her switch pointing to a less preferred one first (CTC step c-iv). This is because such agents do not provide any connection for the cycle and agents on the cycle have incentives to disqualify them to eliminate path overlaps. If S consists only of agents on the cycle, we find the agent i whose path covers another agent j's path to her pointing and let i switch her preference (CTC step c-v). This is because, to ensure everyone has an exclusive path to her pointing, i and j can never hold their pointing together. Agent i should switch preference first because her path contains j's outgoing edges (j's report has impacts on i).

We present an example to illustrate the execution of CTC in Figure 8. In (a), everyone points to her favorite item. Since agent 1 has the minimum order, start from agent 1 and detect cycle $C_5 = \{1, 5\}$. Then, construct the minimum connected component $T = \{1, 2, 4, 5\}$ in (b). By the Path Detection algorithm, we can see that both agent 2 and agent 5 do not have an exclusive path to their pointing, so $S = \{2, 5\}$. By CTC step c-iv, let agent 2 switch her pointing to a less favorite one. In (c), we also start from agent 1 and detect the same cycle C_5, and the same T. Now, only agent 5 does not have an exclusive path to her pointing, so $S = \{5\}$. By CTC step c-v, let agent 5 switch her pointing. In (d), detect cycle $C_3 = \{1, 5, 3\}$ and construct $T = \{1, 2, 3, 4, 5\}$. Everyone has an exclusive path to reach her pointing, so $S = \emptyset$. By CTC step c-iii, we start from agent 2's pointing and find that agent 5 is the last agent whose path to her pointing passes at least one agent in $T \setminus C_3$. So, let agent 5 switch her pointing. Finally in (e), $C_2 = \{1, 5, 4, 2\}$ is a connected cycle and $S = \emptyset$. By CTC step c-ii, C_2 can get traded and agent 3 will switch her pointing as shown in (f). To our best knowledge, the Connected Trading Cycles is the first mechanism that satisfies optimal-cc.



(a) (b) (c) (d) (e) (f)

Figure 8: A running example of CTC. The red arrow is agents' favorite pointing. The dashed red arrow means the agent switch her pointing to a less favorite one. The dashed box shows the minimum connected component T.

Theorem 5.7. *For any order O, CTC is optimal-cc.*

Proof. For any given type profile θ, if the allocation $\pi(\theta)$ given by CTC violates optimal-cc, there exists another allocation $\pi'(\theta)$ such that every agent in a complete component B in $G(\theta)$ is strictly better, while others receive the same allocation. Let i, j be two agents in B, and $\pi'_i(\theta) = \pi_j(\theta) = h_k$. Because B is a complete component in $G(\theta)$, i can reach j by her outgoing edge, which means $i \longrightarrow j$ is an exclusive path for i. Since $\pi(\theta)$ is given by CTC, j has an exclusive path to agent k, the owner of her allocation $\pi_j(\theta)$ (as in CTC step c.ii). Combining the two paths together, agent i can reach k, the owner of $\pi'_i(\theta)$, by $i \longrightarrow j \longrightarrow k$. Hence, if i points to k, CTC will not make agent i switch her pointing to a less preferred agents. Therefore, there exists an exclusive path for every agent $i \in B$ to reach the owner of $\pi'_i(\theta)$, and CTC will allocate $\pi'(\theta)$. This means $\pi(\theta) = \pi'(\theta)$, which contradicts our assumption. □

ACKNOWLEDGMENTS

We thanks the reviewers for their valuable comments on the original submission. This work was supported in part by Science and Technology Commission of Shanghai Municipality (No.22ZR1442200 and No.23010503000), and Shanghai Frontiers Science Center of Human-centered Artificial Intelligence (ShangHAI).

REFERENCES

[1] Atila Abdulkadiroğlu and Tayfun Sönmez. 1999. House allocation with existing tenants. *Journal of Economic Theory* 88, 2 (1999), 233–260.

[2] Atila Abdulkadiroglu and Tayfun Sönmez. 2013. Matching markets: Theory and practice. *Advances in Economics and Econometrics* 1 (2013), 3–47.

[3] David J. Abraham, Katarína Cechlárová, David F. Manlove, and Kurt Mehlhorn. 2005. Pareto Optimality in House Allocation Problems. In *Algorithms and Computation*, Rudolf Fleischer and Gerhard Trippen (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 3–15.

[4] Felix Brandt and Anaëlle Wilczynski. 2019. On the Convergence of Swap Dynamics to Pareto-Optimal Matchings. In *Web and Internet Economics*, Ioannis Caragiannis, Vahab Mirrokni, and Evdokia Nikolova (Eds.). Springer International Publishing, Cham, 100–113.

[5] Sung-Ho Cho, Taiki Todo, and Makoto Yokoo. 2022. Two-Sided Matching over Social Networks. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, Lud De Raedt (Ed.). International Joint Conferences on Artificial Intelligence Organization, 186–193. https://doi.org/10.24963/ijcai.2022/27 Main Track.

[6] Rudolf Fleischer and Yihui Wang. 2008. Dynamic Pareto Optimal Matching. In *Proceedings of the 2008 International Symposium on Information Science and Engieering - Volume 02 (ISISE '08)*. IEEE Computer Society, USA, 797–802. https://doi.org/10.1109/ISISE.2008.237

[7] Laurent Gourvès, Julien Lesca, and Anaëlle Wilczynski. 2017. Object Allocation via Swaps along a Social Network. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence* (Melbourne, Australia) *(IJCAI'17)*. AAAI Press, 213–219.

[8] Martin Hoefer. 2013. Local matching dynamics in social networks. *Information and Computation* 222 (2013), 20–35. https://doi.org/10.1016/j.ic.2012.10.005 38th International Colloquium on Automata, Languages and Programming (ICALP 2011).

[9] Takehiro Kawasaki, Ryoji Wada, Taiki Todo, and Makoto Yokoo. 2021. Mechanism Design for Housing Markets over Social Networks. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems* (Virtual Event, United Kingdom) *(AAMAS '21)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 692–700.

[10] Rajnish Kumar, Kriti Manocha, and Josué Ortega. 2022. On the integration of Shapley–Scarf markets. *Journal of Mathematical Economics* 100 (2022), 102637.

[11] Bin Li, Dong Hao, Hui Gao, and Dengji Zhao. 2022. Diffusion auction design. *Artificial Intelligence* 303 (2022), 103631.

[12] Bin Li, Dong Hao, Dengji Zhao, and Tao Zhou. 2017. Mechanism Design in Social Networks. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence* (San Francisco, California, USA) *(AAAI'17)*. AAAI Press, 586–592.

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

[14] Lloyd Shapley and Herbert Scarf. 1974. On cores and indivisibility. *Journal of mathematical economics* 1, 1 (1974), 23–37.

[15] Bo You, Ludwig Dierks, Taiki Todo, Minming Li, and Makoto Yokoo. 2022. Strategy-Proof House Allocation with Existing Tenants over Social Networks. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems* (Virtual Event, New Zealand) *(AAMAS '22)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1446–1454.

[16] Yao Zhang and Dengji Zhao. 2022. Incentives to Invite Others to Form Larger Coalitions. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems* (Virtual Event, New Zealand) *(AAMAS '22)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1509–1517.

[17] Dengji Zhao. 2021. Mechanism Design Powered by Social Interactions. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems* (Virtual Event, United Kingdom) *(AAMAS '21)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 63–67.